



Universidad
Carlos III de Madrid

PROYECTO FIN DE CARRERA

Aplicación para habilitar conversaciones alrededor de elementos multimedia en redes sociales.

Autor: Miguel Ángel Moreno Armenteros

Tutor: Pablo Alejandro Acuña Ruano

Leganés, noviembre de 2013



(Esta página se ha dejado en blanco intencionadamente)



Título: Aplicación para habilitar conversaciones alrededor de elementos multimedia en redes sociales.

Autor: Miguel Ángel Moreno Armenteros

Director: Pablo Alejandro Acuña Ruano

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día _____ de _____ de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Me gustaría aprovechar este momento para agradecerle a mis padres sus esfuerzos por darme una educación y enseñarme a ser constante y responsable y animarme a continuar cuando el estrés se acumulaba.

A mis compañeros de clase como Miriam, Raquel, Marcos, Jesús, Rafa y muchos otros que me olvido por ser buenos compañeros y mejores amigos.

A los profesores de la UC3M por exigirnos tanto tan constantemente, lo que se apreciaba una vez que te adentras en el mundo laboral.

A mis compañeros de mi primer trabajo como programador en Imaweb, por enseñarme tanto y tan bien y hacerme ver la importancia de hacer las cosas bien en cada línea.

A todos aquellos que un día me dijeron o pensaron que nunca terminaría la ingeniería, que no valía para esto y que hoy en día me preguntan cómo hacer las cosas.

Pero sobre todo me gustaría agradecerle todo su apoyo a Anca, por estar a mi lado durante todo este tiempo, apoyándome sin condiciones y dándome collejas cuando me iba saliendo del camino. Sin ella a mi lado no habría sido posible que hoy escribiera estas líneas.

Gracias a todos, pues en una u otra manera me habéis ayudado a llegar a este momento.

Resumen

Esta aplicación pretende cubrir una necesidad cada vez más creciente: la de ayudar a la comunicación entre diferentes usuarios que no necesariamente se encuentran en la misma ubicación física y que desean o necesitan compartir pensamientos, opiniones, dudas y respuestas sobre uno o varios materiales multimedia.

La finalidad de este proyecto es la de proveer una aplicación accesible desde dispositivos móviles, más concretamente tabletas, que permita a los usuarios de la misma poder ver una colección de varios materiales multimedia de diferente tipo a la vez que interactúan entre ellos mediante el uso de las redes sociales, pudiendo compartir sus impresiones sobre los materiales al mismo tiempo que los visualizan.

Palabras clave: Dispositivo móvil, plataforma móvil, Smartphone, Tableta, Android, iOS, colaborativo, redes sociales, Twitter, Multimedia, interacción.

Abstract

This application intends to cover an increasing need: Helping different users that are not necessarily in the same physical location but need or want to share thoughts, ideas, opinions, questions and answers over one or more multimedia materials.

The purpose of this project is providing one application which will be accessible from mobile devices – in particular, tablets – that will allow its users to view a set of multimedia materials of different kinds at the same time that they interact by the use of the social networks, being able to share their personal impressions about these multimedia materials at the time they visualize them.

Keywords: Mobile device, mobile platform, Smartphone, Tablet, Android, iOS, collaborative, social networks, Twitter, Multimedia, interaction.



(Esta página se ha dejado en blanco intencionadamente)

Tabla de contenido

Índice de Tablas.....	10
Índice de Figuras.....	12
1. Introducción.....	14
1.1 Objetivos	14
1.2 Estructura del documento	14
2. Estado del Arte	16
2.1 Aplicaciones existentes.....	17
2.1.1 Google Hangouts	17
2.1.2 Twitter	18
2.1.3 Slideshare	19
2.1.4 Lanyrd.com.....	19
2.1.5 Eventmobi.....	20
2.1.6 StageIt.....	21
2.1.7 Comparativa de las aplicaciones encontradas.....	22
2.2 Tecnologías disponibles consideradas para el desarrollo de la solución	23
2.3 Desarrollo de una solución mediante una Web HTML5 optimizada para tabletas	23
2.3.1 HTML	23
2.3.2 CSS	27
2.3.3 Javascript	30
2.3.4 Compatibilidad de HTML5 entre los diferentes navegadores de terminales móviles.....	31
2.4 Desarrollo de una solución mediante una aplicación nativa en una de las plataformas móviles más populares.....	35
2.4.1 Windows Phone.....	36
2.4.2 iOS.....	39
2.4.3 Android	42
3. Desarrollo.....	60
3.1 Definición del problema	60
3.2 Metodología a seguir en el desarrollo del proyecto	60
3.2.1 Metodología ágil en el desarrollo de software	60
3.3 Desarrollo del proyecto	62
3.4 Primera iteración.....	64

3.4.1	Metas generales.....	64
3.4.2	Toma de requisitos.....	64
3.4.3	Identificación de los Requisitos	64
3.4.4	Casos de uso	77
3.4.5	Wireframes.....	79
3.4.6	Implementación	90
3.5	Segunda Iteración.....	100
3.5.1	Metas Generales	100
3.5.2	Toma de Requisitos.....	101
3.5.3	Casos de uso de la aplicación	103
3.5.4	Wireframes.....	104
3.5.5	Implementación	105
3.6	Resultado Final de la aplicación	106
3.6.1	Pantalla inicial de la aplicación	106
3.6.2	Pantalla de opciones.....	106
3.6.3	Escenario A: Creación de evento.....	107
3.6.4	Escenario B: Visualización de un evento	112
4.	Planificación	116
5.	Presupuesto.....	118
5.1	Costes de personal	118
5.2	Costes de material.....	118
5.2.1	Costes de hardware.....	119
5.2.2	Costes de software	119
5.2.3	Coste total del material.....	120
5.3	Coste total del proyecto	120
6.	Conclusiones y líneas futuras.....	121
6.1	Conclusiones.....	121
6.2	Líneas futuras	122
7.	Conclusiones personales.....	125
8.	Acrónimos	126
9.	Glosario de términos.....	127
10.	Referencias.....	130
11.	ANEXO A.....	134
	Imágenes de los <i>wireframes</i> iniciales desarrollados a mano	134

Índice de Tablas

Tabla 1. Comparativa entre aplicaciones encontradas, parte A.....	22
Tabla 2. Comparativa entre aplicaciones encontradas, parte B.....	22
Tabla 3. Principales modificaciones en las etiquetas entre HTML4 y HTML5	25
Tabla 4. Nuevos tipos de “input” en HTML5.....	26
Tabla 5. Selectores CSS básicos	29
Tabla 6. Resumen de Android Cupcake	43
Tabla 7. Resumen de Android Donut	43
Tabla 8. Resumen de Android Eclair	43
Tabla 9. Resumen de Android Froyo	43
Tabla 10. Resumen de Android Gingerbread	44
Tabla 11. Resumen de Android Honeycomb	44
Tabla 12. Resumen de Android Ice Cream Sandwich	44
Tabla 13. Resumen de Android Jelly Bean	45
Tabla 14. Cuota de mercado de las diferentes versiones Android (abril '13)	46
Tabla 15. Comparativa Windows Phone-iOS-Android	57
Tabla 16: Plantilla de requisitos del sistema.....	62
Tabla 17: Requisito de software funcional 01	64
Tabla 18: Requisito de software funcional 20	65
Tabla 19. Requisito de software funcional 30	65
Tabla 20: Requisito de software funcional 40	65
Tabla 21: Requisito de software funcional 50	65
Tabla 22: Requisito de software funcional 60	66
Tabla 23: Requisito de software funcional 70	66
Tabla 24: Requisito de software funcional 80	67
Tabla 25: Requisito de software funcional 90	67
Tabla 26: Requisito de software funcional 100.....	67
Tabla 27: Requisito de software funcional 110.....	67
Tabla 28: Requisito de software funcional 120.....	68
Tabla 29: Requisito de software funcional 130.....	68
Tabla 30: Requisito de Software no funcional de comprobación 10	69
Tabla 31: Requisito de Software no funcional de comprobación 20	69
Tabla 32: Requisito de Software no funcional de comprobación 30	69
Tabla 33: Requisito de Software no funcional de comprobación 40	69
Tabla 34: Requisito de Software no funcional de comprobación 50	70
Tabla 35: Requisito de Software no funcional de comprobación 60	70
Tabla 36: Requisito de Software no funcional de comprobación 70	70
Tabla 37: Requisito de Software no funcional de comprobación 80	70
Tabla 38: Requisito de Software no funcional de comprobación 90	71
Tabla 39: Requisito de Software no funcional de comprobación 100.....	71
Tabla 40: Requisito de Software no funcional de comprobación 110.....	71
Tabla 41: Requisito de Software no funcional de comprobación 120.....	71
Tabla 42: Requisito de Software no funcional de comprobación 130.....	72
Tabla 43: Requisito de Software no funcional de comprobación 140.....	72
Tabla 44: Requisito de Software no funcional de comprobación 150.....	72

Tabla 45: Requisito de Software no funcional de comprobación 160.....	72
Tabla 46: Requisito de Software no funcional de comprobación 170.....	73
Tabla 47: Requisito de Software no funcional de comprobación 180.....	73
Tabla 48: Requisito de Software no funcional de comprobación 190.....	73
Tabla 49: Requisito de Software no funcional de comprobación 200.....	73
Tabla 50: Requisito de Software no funcional de restricción 10.....	74
Tabla 51: Requisito de Software no funcional de restricción 20.....	74
Tabla 52: Requisito de Software no funcional de restricción 30.....	74
Tabla 53: Requisito de Software no funcional de restricción 40.....	74
Tabla 54: Requisito de Software no funcional de restricción 50.....	75
Tabla 55: Requisito de Software no funcional de interfaz de usuario 10	75
Tabla 56: Requisito de Software no funcional de interfaz de usuario 20	75
Tabla 57: Requisito de Software no funcional de interfaz de usuario 30	75
Tabla 58: Requisito de Software no funcional de interfaz de usuario 40	76
Tabla 59: Requisito de Software no funcional de interfaz de usuario 50	76
Tabla 60: Requisito de Software no funcional de interfaz de usuario 60	76
Tabla 61. Ejemplo de información sobre un tuit que devuelve la API de Twitter	94
Tabla 62. Ejemplo de respuesta a una petición oEmbed en youtube	95
Tabla 63. Resumen del servicio para verificar la existencia de un hashtag.....	96
Tabla 64. Resumen del servicio para registrar un nuevo evento.....	97
Tabla 65. Resumen del servicio para registrar un nuevo material en un evento.	97
Tabla 66. Resumen del servicio para obtener la información de un evento.....	98
Tabla 67: Requisito de Software no funcional de comprobación 100 eliminado	
.....	101
Tabla 68: Requisito de Software no funcional de interfaz de usuario 70	102
Tabla 69: Requisito de Software no funcional de interfaz de usuario 80	102
Tabla 70: Requisito de Software no funcional de interfaz de usuario 90	102
Tabla 71: Requisito de Software no funcional de interfaz de usuario 100.....	103
Tabla 72: Requisito de Software no funcional de interfaz de usuario 110.....	103
Tabla 73: Requisito de Software no funcional de interfaz de usuario 110.....	103
Tabla 74. Resumen de costes de personal del proyecto	118
Tabla 75. Costes de hardware del proyecto	119
Tabla 76. Costes de software del proyecto	119
Tabla 77. Resumen de gastos de material atribuibles al proyecto	120
Tabla 78. Coste total del proyecto sin 21% IVA	120
Tabla 79. Evaluación de la aplicación desarrollada.....	121

Índice de Figuras

Figura 1. Logo de Google Hangouts.....	17
Figura 2. Captura de pantalla de la interfaz de Twitter para iPad donde un usuario comparte un video	18
Figura 3. Implementaciones del API de Slideshare disponibles con fecha 9 de febrero de 2013	19
Figura 4. Captura de pantalla de un evento en lanyrd.com	20
Figura 5. Información sobre el evento ofrecida por eventmobi	21
Figura 6. Captura de pantalla de una aplicación Web móvil que utiliza jQuery Mobile.....	32
Figura 7. Resultados de “caniuse.com” para la funcionalidad HTML5 Websockets	33
Figura 8. Ejemplo de la tabla con los datos que nos ofrece “mobilehtml5.org”	33
Figura 9. Penetración de Smartphones en EE.UU[URL20]	35
Figura 10. Distribución de SO Móviles en EE.UU[URL22]	36
Figura 11. Arquitectura Windows Phone	37
Figura 12. Arquitectura por capas de iOS.....	40
Figura 13. Logo de Android Cupcake	43
Figura 14. Logo de Android Donut.....	43
Figura 15. Logo de Android Eclair.....	43
Figura 16. Logo de Android FroYo	43
Figura 17. Logo de Android Gingerbread	44
Figura 18. Logo de Android Honeycomb	44
Figura 19. Logo de Android Ice cream sándwich	44
Figura 20. Logo de Android Jelly Bean.....	45
Figura 21. Distribución de la cuota de mercado entre las diferentes versiones ..	46
Figura 22. Pila software de Android.....	47
Figura 23. Arquitectura Android.....	47
Figura 24. Ciclo de vida de una actividad en Android	51
Figura 25. Representación gráfica del desarrollo ágil de software	61
Figura 26. Caso de uso para usuario no logueado.....	77
Figura 27. Caso de uso para usuario logueado	77
Figura 28. Caso de uso para usuario asistiendo a evento.....	78
Figura 29. Caso de uso para usuario creando un evento.....	78
Figura 30. Boceto de imagen inicial de la aplicación para autorizar el uso de la cuenta de Twitter del usuario	79
Figura 31. Boceto de la pantalla de opciones de la aplicación.....	80
Figura 32. Boceto del primer paso en la creación de eventos	81
Figura 33. Boceto del proceso de selección de fechas durante la creación de un evento	81
Figura 34. Boceto de feedback al usuario mientras se validan los datos	82
Figura 35. Boceto de error cuando el hashtag ya ha sido usado en la aplicación ..	82
Figura 36. Boceto de error porque el hashtag no se ha introducido	83
Figura 37. Boceto del paso 2 de la creación de eventos.....	83

Figura 38. Boceto de información de error al usuario porque no ha etiquetado el evento	84
Figura 39. Boceto de información de error al usuario porque no ha facilitado los usuarios a notificar con un formato correcto	84
Figura 40. Boceto de notificación de error cuando el usuario no pone una descripción de evento	85
Figura 41. Boceto de notificación de error cuando el usuario pone una descripción de evento demasiado corta.....	85
Figura 42. Boceto del Paso 3 de creación de evento	86
Figura 43. Boceto de notificación de error porque no se han introducido materiales	86
Figura 44. Boceto de pantalla de adición de nuevo material	87
Figura 45. Boceto de error en el formato de la URL	87
Figura 46. Boceto de notificación de evento creado con éxito	88
Figura 47. Boceto de visualización de evento	88
Figura 48. Boceto de visualización de evento con menú lateral desplegado.....	89
Figura 49. Boceto de error en el segundo paso de la creación de evento reeditado, descripción vacía.....	104
Figura 50. Boceto de error en el segundo paso de la creación de evento reeditado, descripción demasiado corta	104
Figura 51. Pantalla inicial de la aplicación	106
Figura 52. Pantalla de opciones.....	107
Figura 53. Paso 1 de la creación de un evento.....	107
Figura 54. Teclado virtual de Android sobre el paso 1	108
Figura 55. Calendario para la introducción de fechas	108
Figura 56. Primer paso debidamente cumplimentado.....	109
Figura 57. Segundo paso de creación de evento sin completar	109
Figura 58. Segundo paso de creación de evento completado	110
Figura 59. Tercer paso de la creación de evento	110
Figura 60. Notificación de error mediante ventana emergente.....	111
Figura 61. Pantalla para añadir un nuevo material multimedia.....	111
Figura 62. Error durante la creación de un nuevo material para un evento	112
Figura 63. Datos generales del evento.....	113
Figura 64. Visualización de una imagen.....	113
Figura 65. Visualización de un video de Youtube.....	114
Figura 66. Visualización de una presentación Slideshare	114
Figura 67. El menú deslizante de la aplicación activo	115
Figura 68. Resumen de la planificación mostrando fechas de inicio y fin de cada tarea y su esfuerzo.....	116
Figura 69. Diagrama de Gantt representativo de la planificación del proyecto .	117

1.Introducción

Los dispositivos móviles tienen un papel muy importante en la sociedad actual, siendo éstos una plataforma ideal para la comunicación interpersonal, permitiéndonos hablar con otro ser humano que se encuentra a varios kilómetros de distancia ya sea por voz o por texto e incluso transmitirle archivos como fotografías, pero desde la aparición de los dispositivos inteligentes, su uso va más allá, permitiéndonos verificar el correo electrónico, acceder a internet, seguir mapas, obtener indicaciones para llegar a un destino, escuchar música, grabar videos, tomar fotografías, interactuar con nuestros amigos en las redes sociales o consultar nuestro expediente académico o saldo del banco.

Se busca desarrollar una aplicación que sea accesible desde tabletas de manera que se puedan presentar diferentes tipos de materiales multimedia al mismo tiempo que distintos usuarios de la aplicación puedan interactuar entre ellos para poder comentar sobre dichos materiales.

Esta aplicación pretende cubrir una necesidad cada vez más creciente: la de ayudar a la comunicación entre diferentes usuarios que no necesariamente se encuentran en la misma ubicación física y que desean o necesitan compartir pensamientos, opiniones, dudas y respuestas sobre uno o varios materiales multimedia.

1.1 Objetivos

El objetivo principal de este proyecto es proporcionar una plataforma que permita reunir a gente para generar conversaciones mediante el uso de las redes sociales sobre uno o varios materiales multimedia.

Los objetivos secundarios son, a su vez:

- Permitir a los usuarios la creación de salas de conversación que referencien diferentes materiales multimedia.
- Utilizar Twitter como red social para conversar sobre los materiales
- Utilizar en la aplicación al menos los siguientes tipos de materiales multimedia alojados en internet: Pases de diapositivas, imágenes y videos.

1.2 Estructura del documento

Este documento presenta la siguiente estructura:

En el **capítulo 2** se tratará el **estado del arte**, donde se dará a conocer qué aplicaciones se han encontrado a la fecha de escritura de este documento.

En el **capítulo 3** se darán a conocer los detalles de cómo se ha hecho el **desarrollo**, tanto la metodología utilizada como el proceso que se ha seguido

para elaborar la aplicación, incluyendo los diseños y los detalles de implementación.

En el **capítulo 4** se muestra la **planificación del proyecto** que se ha seguido.

En el **capítulo 5** se detalla el **presupuesto del proyecto**, desglosando los gastos de hardware y software.

En el **capítulo 6** se describen las **conclusiones** obtenidas tras la realización del proyecto y las **líneas futuras** que se pueden seguir si se continúa el desarrollo.

En el **capítulo 7** se dan a conocer las **conclusiones personales**.

En el **capítulo 8** se definen los **acrónimos** no especificados en el documento y que pueden ser de interés al lector.

En el **capítulo 9** se desarrolla un **glosario de términos** que pueden ser útiles para entender el documento pero cuya definición no procede durante la lectura del mismo.

En el **capítulo 10** se presentan las **referencias** consultadas para la elaboración del documento.

En el **capítulo 11** se pueden encontrar como un **anexo** unas capturas de los **diseños iniciales** de la aplicación.

2.Estado del Arte

Hoy en día se pueden encontrar multitud de aplicaciones (mayoritariamente Web) que ofrecen un material multimedia determinado como pueden ser videos (en directo o en diferido), clips de sonido, pases de diapositivas o documentos de texto, entre otros, y que permiten comentar sobre dicho material. Sin embargo, los comentarios pueden ser únicamente sobre ese contenido y únicamente sobre ese tipo determinado de contenido (pues es el único que ofrecen).

Por norma general, cuando alguien desea compartir un contenido con otros usuarios y comentar sobre él, comparte el enlace que lleva a dicho contenido, haciendo difícil mantener una conversación alrededor del mismo mientras se visiona fuera de su Web de origen. Así mismo, a la hora de compartir materiales multimedia, suele hacerse de uno en uno, no se han encontrado hasta la fecha aplicaciones que permitan agrupar distintos materiales multimedia y a la vez permitan a los diferentes usuarios de la misma interactuar entre ellos de manera fluida..

Otra parte fundamental para el proyecto es que la interacción entre los diferentes usuarios se lleve a cabo mediante el uso de las redes sociales. Hoy día, las redes sociales tienen un gran impacto en la sociedad y sirven como medio para conversar con gente conocida y desconocida entre sí sobre temas de diversa índole, como un partido de fútbol, un programa de televisión, o un hallazgo científico.

Así pues, el objetivo de este proyecto es proporcionar una plataforma que permita reunir a gente para generar conversaciones mediante el uso de las redes sociales (en nuestro caso Twitter) sobre uno o varios materiales multimedia, ya sea un video online (como por ejemplo un capítulo de alguna serie gratuita a través de Internet) o una colección de videos, fotos o pases de diapositivas. De este modo, además de modo lúdico, podría utilizarse como apoyo a conferencias o en el ámbito del e-learning para los MOOCs (Massive Online Open Courses - que en inglés español significaría algo como “Cursos Masivos de Acceso Online y Gratuito”).

2.1 Aplicaciones existentes

Tras realizar una investigación por Internet y los diferentes repositorios de aplicaciones de plataformas móviles, se han encontrado aplicaciones y páginas Web que cumplen parcialmente los requerimientos de nuestro proyecto, como los ejemplos siguientes:

- Google Hangouts (<http://www.google.com/hangouts>)
- Twitter (<http://www.twitter.com>)
- Slideshare (<http://www.slideshare.net>)
- Lanyrd (<http://lanyrd.com>)
- Eventmobi (<http://www.eventmobi.com>)
- StageIt (<http://www.stageit.com>)

2.1.1 Google Hangouts

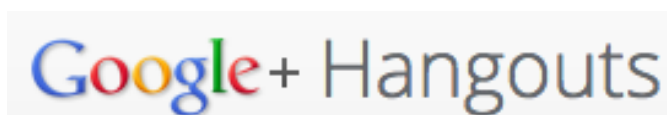


Figura 1. Logo de Google Hangouts

Google Hangouts es una herramienta de Google estrechamente ligada al resto de la plataforma (Google+, Youtube, Google Docs...) que permite realizar videoconferencias de hasta 10 participantes con audio y video de manera gratuita o de hasta 15 participantes para empresas, gobiernos o escuelas que utilicen Google Apps [URL 1]. Adicionalmente, permite compartir el escritorio y documentos de Google Docs con el resto de usuarios y con la opción Hangouts On Air, permite grabar video y emitirlo en directo al mismo tiempo a través de Google+ y nuestra cuenta de Youtube.

Aunque la herramienta es muy potente, solamente permite tener 10 o 15 participantes de forma simultánea a la mayoría de usuarios. Para entablar una conversación ("*Hangout*"), todos los participantes han de estar conectados al mismo tiempo y el contenido no necesariamente perdura para posteriores visitas aunque se puede guardar para verse en Youtube.

Por tanto, aún siendo una herramienta muy potente que permite compartir cualquier material a partir de compartir el escritorio y, por ende, cualquier material que pueda mostrarse en un ordenador. Esto se hace en forma de video, por lo que cualquier participación externa a los participantes del Hangout o posterior al mismo no es posible – es típico que se utilice Twitter como herramienta de apoyo para que más usuarios participen o hagan llegar sus dudas y comentarios. Esto hace que no cumpla los requisitos de nuestro problema.

2.1.2 Twitter

Tanto la versión Web como la aplicación oficial para dispositivos móviles y tantos otros clientes que se han hecho, permiten interactuar con nuestro perfil de la red social Twitter, lo que implica la posibilidad de compartir imágenes, videos de Internet o incluso, mediante *twitcam* por ejemplo, compartir video en directo. Sin embargo, lo que se comparte es un enlace al recurso (foto, página Web...), lo que dificulta ver el contenido a la vez que se comenta sobre él. En el caso de las imágenes y enlaces de YouTube, se está mejorando la interfaz y ya son muchos los clientes que permiten verlos al tiempo que se lee el tuit, como se ve en la figura 3.

Además, las conversaciones múltiples no son cómodas y necesitan contener el nombre de usuario de los participantes junto al enlace, lo que unido al máximo de 140 caracteres por mensaje, hacen que la experiencia no sea todo lo cómoda que debería.

Por otra parte, Twitter ofrece una poderosa API REST para desarrolladores, de manera que éstos pueden acceder a la información pública que ofrezca la plataforma y crear sus propios clientes o incluso usar las credenciales de usuario de Twitter para permitir el acceso a usuarios en una aplicación.

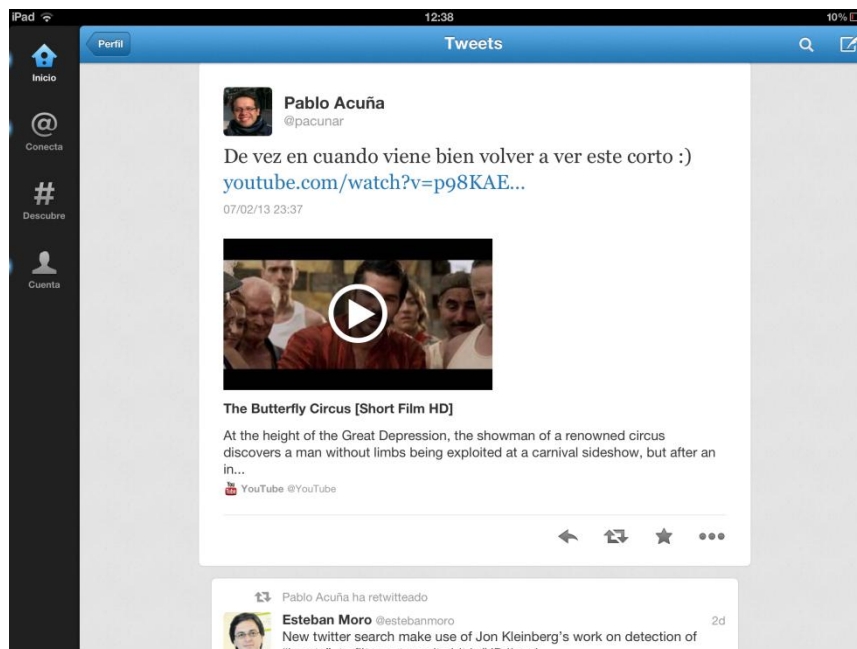


Figura 2. Captura de pantalla de la interfaz de Twitter para iPad donde un usuario comparte un video

2.1.3 Slideshare

Slideshare es una aplicación Web que permite subir nuestras presentaciones de diapositivas para poder utilizarlas desde cualquier parte del mundo y a la vez poder compartirlas con otras personas, comentarlas, etcétera. Aunque técnicamente es contenido multimedia mezclado con la interacción entre los usuarios, se restringe a un único tipo de contenido multimedia: el pase de diapositivas. Además, solamente cuenta con versión Web, aunque tiene una versión optimizada para acceso desde dispositivos móviles.

Posee un API REST para desarrolladores que permite acceder a sus recursos a través de Internet, además de tener librerías implementadas en los lenguajes de desarrollo Web más habituales [URL2], como se observa en la figura 4.

Aunque nació en 2006, poco a poco se fue abriendo camino y hoy en día es bastante frecuente acudir a una clase, conferencia o un evento de cualquier tipo y alojar previa o posteriormente las presentaciones en Slideshare.

Developers & API	SlideShare API » API Kits		
Documentation			
Apply for API key			
API Explorer			
Resources			
Developer Forum			
API Kits			
Featured Apps			
Terms of Service			
oEmbed			
Player API			
Developer Roadmap			
	Platform	Name	Contributor
	PHP	SSUtil	SlideShare
	Microsoft .net (C#)	DotNetUtil	Frederik Vig
	Java	JSlideShare	Lucio Benfante
	Python	PySlideShare	Subramanian Prabhu
	Python	MediaWiki Widget	Sergey Chernyshev
	Coldfusion	SlideshareCFC	Raymond Camden
	PERL	PERL Wrapper	Ashish Mukherjee
	Note: The API kits are not maintained by Slideshare and should be used at your own risk. For API kit support, please join the developers mailing list or contact the maintainers directly.		
	We are looking for more API wrappers in cURL, Perl, ActionScript, ColdFusion, Ruby, VB.NET, J# and other languages. Please submit your wrapper at api@slideshare.net and we'll feature it on this section.		

Figura 3. Implementaciones del API de Slideshare disponibles con fecha 9 de febrero de 2013

2.1.4 Lanyrd.com

Lanyrd es una plataforma Web de apoyo a conferencias y eventos que, normalmente, se desarrollan físicamente en un determinado lugar. Su único objetivo es el de almacenar en un repositorio centralizado la información sobre conferencias, sus conferenciantes y asistentes. De esta manera, permite indicar quiénes serán los conferenciantes, quienes dicen que asistirán y quienes han decidido seguir información sobre el evento (*track*) porque aún no es seguro que puedan asistir. Como se puede apreciar en la figura 4, la interfaz es poco compleja.



Figura 4. Captura de pantalla de un evento en lanyrd.com

Permite añadir información relativa a un evento, ya sean imágenes del evento, referencias a libros que traten la temática del evento, referencias a la compra de tickets, etc., pero no es suficiente para cumplir los requisitos de nuestra aplicación.

2.1.5 Eventmobi

Eventmobi es una compañía con base en Toronto, Canadá, que desarrolla una aplicación Web para móvil bajo pago desarrollada en HTML5[URL3] que tiene el objetivo de servir de ayuda en eventos y conferencias de todo tipo. Su uso abarca desde lo que llaman el “pre-evento” hasta el “post-evento”. En el “pre-evento” permite configurar la aplicación con el logo y colores de empresa, vender espacio publicitario a “sponsors” e incluso ayuda con el material publicitario mediante códigos QR, etc.

En la fase 2, “on-site”, es decir, cuando la aplicación está accesible al público general, ofrece muchas características. Los visitantes del evento pueden tomar notas, marcar temas de interés, tener planos interactivos del complejo donde se lleve a cabo el evento, acceso a documentos, etc.

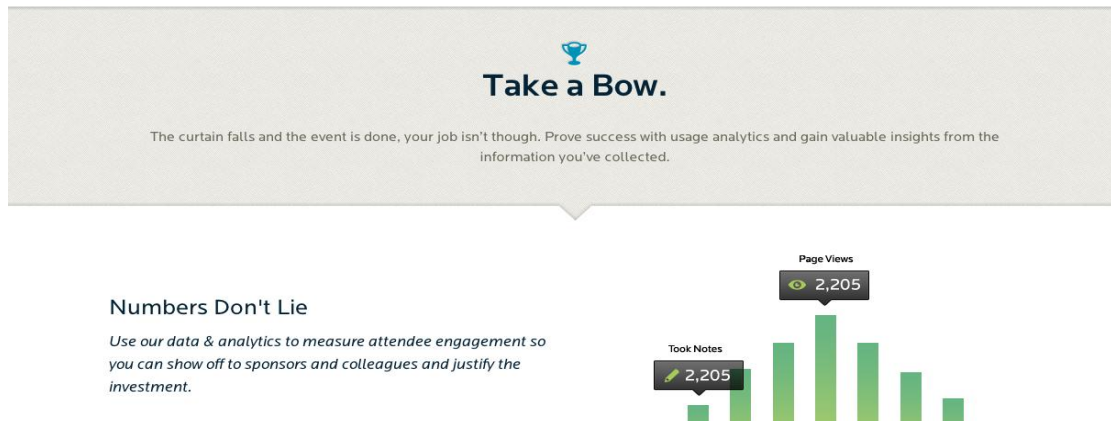


Figura 5. Información sobre el evento ofrecida por eventmobi

La fase final o “post-evento”, le muestra al organizador estadísticas sobre el uso de la aplicación para el evento.

Aunque la aplicación es bastante completa y configurable, es una aplicación de apoyo a eventos que tienen lugar físicamente, por lo que no está pensada para que los usuarios se comuniquen entre ellos al mismo tiempo que acceden a contenido multimedia y hagan aportaciones sobre el contenido visualizado con los demás asistentes al evento o para eventos virtuales.

2.1.6 Stagelt

Stagelt es una plataforma de conciertos virtual que permite conectar a artistas de todo el mundo con el público simplemente utilizando su ordenador personal. El artista puede crear el evento y a través de la página, vender entradas. Quienes compran entradas podrán interactuar entre ellos mismos gracias a un chat y con el artista a través de la videocámara. Además, existe un bote de propinas donde los usuarios pueden premiar la actuación del artista.

Al igual que sucede con otras aplicaciones existentes analizadas, no cumple la totalidad de los requisitos, ya que aunque permite a los usuarios interactuar entre ellos mientras visualizan un contenido multimedia, éste está ligado al video en directo. Además, todos los participantes han de coincidir al mismo tiempo que se desarrolla el evento para poder participar.

2.1.7 Comparativa de las aplicaciones encontradas

	Google Hangouts	Lanyrd	Twitter
Chat entre asistentes	Texto, Video	No	Sí, mediante menciones (@usuario)
Visualización del multi-media Online	Sí	No	Depende del cliente utilizado
Tipos de multimedia ofrecido	Video	Enlaces a videos, audios y pases de diapositivas, todo alojado por terceros	Enlaces, imágenes, vídeos
Tiene integración con Redes Sociales	Google+	Twitter, LinkedIn	Es una red social
Participantes simultáneos	10 a 15	No existe interacción	Limitado a las menciones que quepan en 140 caracteres. Además, debe caber el texto en esos 140 caracteres.
Dispone de versión para móvil	Web Móvil, Aplicación Nativa	Web móvil, Aplicación nativa para iOS	Web móvil, Aplicación Nativa

Tabla 1. Comparativa entre aplicaciones encontradas, parte A

	SlideShare	Eventmobi	StageIt
Chat entre asistentes	Mediante el sistema de comentarios	Desconocido	Sí
Visualización del multi-media Online	Sí	Desconocido	Sí
Tipos de multimedia ofrecido	Pase de diapositivas	Desconocido	Video en directo
Tiene integración con Redes Sociales	Facebook, LinkedIn, Twitter, Google+, Pinterest...	Desconocido	Twitter, Facebook
Participantes simultáneos	No hay participantes	Desconocido	Más de 11.000 ¹
Dispone de versión para móvil	Web Móvil	Web Móvil	No

Tabla 2. Comparativa entre aplicaciones encontradas, parte B

¹ Según respuesta de su servicio de soporte al usuario, han llegado a tener más de 11000 personas viendo e interactuando en el mismo concierto.

2.2 Tecnologías disponibles consideradas para el desarrollo de la solución

Para cubrir las necesidades descritas existen varias tecnologías disponibles. Las que se han considerado son:

- Web Móvil con HTML5: Desarrollo de una Web optimizada para su acceso desde tabletas, ayudándonos probablemente de “*Frameworks*” que agilicen el desarrollo de la parte del cliente (o “front-end”) y la parte del servidor (o “back-end”) y utilizando el conjunto de herramientas que ofrece HTML5.
- Desarrollo de una aplicación nativa para una de las plataformas más populares
 - **Windows Phone:** Desarrollo de una aplicación para terminales con software de Microsoft®.
 - **iOS:** Desarrollo de una aplicación para iPad, la tableta de Apple.
 - **Android:** Desarrollo de una aplicación para plataforma Android, optimizada para su instalación en tabletas de al menos 7 pulgadas.

Antes de poder decantarse por una de las opciones, se debe entender qué son, qué ventajas y qué dificultades tienen para el desarrollo de la solución y así poder elegir la opción que mejor resuelva nuestro problema.

2.3 Desarrollo de una solución mediante una Web HTML5 optimizada para tabletas

El desarrollo de una solución mediante el uso de una Web optimizada o pensada para tabletas, requiere de un amplio abanico de tecnologías y lenguajes de maquetado y programación. Para el maquetado / presentación de la información al usuario final, se utiliza el lenguaje de marcado HTML. Para la parte del cliente es casi obligatorio, hoy en día, el uso de JavaScript, y en el caso del uso del entorno HTML5, el uso de CSS3. Para la parte del servidor, donde se desarrolla la lógica de la aplicación que no puede llevarse a cabo en el lado del cliente, hay multitud de lenguajes, con diferente popularidad y complejidad, ventajas e inconvenientes. Se empezará hablando del lado del cliente, y más concretamente, por HTML, para seguir con CSS y Javascript.

2.3.1 HTML

HTML (HyperText Markup Language, por sus siglas en inglés) es un lenguaje de marcado que, mediante el uso de etiquetas de varios tipos, permite determinar la estructura final de un documento. De esta manera permite definir encabezados, títulos, tablas, listas de elementos numeradas y sin numerar, párrafos y enlaces a otros documentos, entre otros.

Poco a poco ha ido evolucionando, añadiendo nuevas etiquetas y dejando algunas obsoletas, hasta llegar a la versión HTML4.01, precursora de HTML5.

HTML5 es un cambio conceptual significativo respecto a la especificación 4.01, que incluye no sólo un cambio en las etiquetas, sino un cambio o integración con las otras tecnologías predominantes en el desarrollo del lado del cliente (conocido también como “*front-end*”) para la Web.

2.3.1.1 Novedades de HTML5

Si se escuchara de varias personas o fuentes una descripción de lo que es HTML5, tal vez cada una de ellas dé una definición diferente. Para unos, HTML5 es solamente una remodelación del lenguaje de marcado HTML, es decir, una evolución de HTML4 en la que se han añadido nuevas etiquetas y se han dado por obsoletas otras cuantas. Para otros, HTML5 incluye no solamente las nuevas etiquetas sino que incluye además una serie de funcionalidades que solamente están presentes en algunos navegadores. Sin embargo, HTML5 es más que eso.

HTML5 viene definido por el consorcio W3C (de sus siglas en inglés: “World Wide Web Consortium”) y éste, hasta la fecha, ha definido más de 100 nuevas especificaciones para la “nueva Web”. Aunque se hayan definido más de 100 nuevas especificaciones - de las cuales se va a hablar posteriormente sobre las que se consideran más importantes o relevantes en un proyecto de estas características - se puede resumir diciendo que HTML5 es un conjunto de especificaciones dadas por el W3C que engloban cambios en las tecnologías HTML, CSS y JavaScript.

2.3.1.2 Consorcio W3C

El consorcio W3C es una organización que forma grupos de consenso para definir y mejorar estándares. Para ello crea comités (llamados “grupos de trabajo”) los cuales están formados por empresas o personas relevantes en el sector. Por ejemplo, Apple, Microsoft, Google y la fundación Mozilla forman parte del grupo de trabajo para HTML5, por lo que dada la variedad de miembros, se pretende conseguir un estándar que no siga la tendencia de un único fabricante y que beneficie a todos los usuarios finales.

2.3.1.3 Principales diferencias de HTML5 y HTML4

Aunque al hablar de HTML5 se hable en realidad de una combinación de 3 tecnologías como son HTML, Javascript y CSS3, HTML5 como lenguaje de marcado incorpora cambios sustanciales en las etiquetas que lo componen, añadiendo unas y modificando o eliminando otras. En el espacio para el aprendizaje del consorcio W3C, W3Schools, se puede encontrar una tabla comparativa que ayuda a identificar de un vistazo las etiquetas añadidas, modificadas y desaparecidas[URL4].

En la Tabla siguiente se puede ver un resumen de la relación de etiquetas de HTML que han variado entre HTML4 y HTML5 y que tienen posible uso en aplicaciones multimedia, donde se han marcado en naranja las etiquetas

modificadas, en rojo las eliminadas y en verde las nuevas incorporaciones. En blanco, las etiquetas no modificadas.

Etiqueta	Atributos	Comentarios
<a>	href target rel hreflang media type	Atributo Añadido: <i>media</i> Atributo cambiado: Target
<applet>		Etiqueta Eliminada
<audio>	autobuffer autoplay controls loop src	Nueva Etiqueta
<canvas>	height width	Nueva Etiqueta
<embed>	height src type width	Nueva Etiqueta
<iframe>	Estándar o ninguno	
	Estándar o ninguno	
<input>	accept alt auto-complete autofocus checked disabled form formaction formenctype formmethod formnovalidate formtarget height list max maxlength min multiple name pattern placeholder readonly required size src step type value width	Etiqueta cambiada: Añadidos 13 elementos a type
<map>	Estándar o ninguno	
<script>	Estándar o ninguno	
<source>	media src type	Nueva Etiqueta
<video>	src poster autobuffer autoplay loop controls width height	Nueva Etiqueta

Tabla 3. Principales modificaciones en las etiquetas entre HTML4 y HTML5

Aunque de forma nativa HTML no ofrece grandes posibilidades, se observa un avance dándole mayor importancia a estas etiquetas que favorecen la inclusión de elementos multimedia importantes como audio y video (nuevas etiquetas) sin necesidad de aplicaciones de terceros que lo ejecuten. Además, gracias a la etiqueta <canvas> se pueden incrustar dibujos en la página utilizándola como lienzo mediante el uso de Javascript, añadido a la página con la etiqueta <script>. Se ha eliminado también la etiqueta para incluir *applets*, los cuales se desarrollaban en Java y han caído en desuso en beneficio de Flash y Javascript.

2.3.1.4 Validación nativa de los elementos de un formulario

En HTML5 se han incluido, a la escritura de este documento (febrero de 2013), al menos trece nuevos tipos de entrada de datos de usuario[URL6], aunque no todos están ampliamente soportados por los principales navegadores. Esta es la lista de los nuevos tipos de campos para obtener datos introducidos por los usuarios. Sin embargo, en la mayoría de casos aún no se soportan o su uso actual es puramente semántico.

Tipo de input	Descripción
Color	Define un selector de colores
Date	Define un selector de fecha
Datetime	Define un selector de fecha con selector de horas, minutos y segundos, basado en zona UTC
datetime-local	Define un selector de fecha con selector de horas, minutos y segundos, sin basarse en zona
Email	Define un campo para insertar un email, validando que tenga un formato válido
Month	Define un campo para introducir mes y año
Number	Define un campo para introducir un número
Range	Define un campo con un selector desplazable para elegir un valor dentro del rango dado
Search	Define un campo para introducir términos a buscar
Tel	Define un campo para introducir un teléfono
time	Define un campo para introducir una hora
url	Define un campo para introducir una URL, validando el formato
week	Define un campo para introducir una semana y año

Tabla 4. Nuevos tipos de “input” en HTML5

Además, también se ha incluido un atributo que marca el campo como requerido (“required”) y un atributo “pattern” (patrón, en inglés) que permite definir una expresión regular que valide el contenido.

Por ejemplo:

El siguiente código solicitaría un campo con formato IPv4 que es obligatorio[URL7]

```
<input type="text" pattern="\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}" name="ipv4" title="Introduzca una IPv4" required>
```

2.3.2 CSS

Como se ha comentado anteriormente, HTML sirve para estructurar el documento, pero no le otorga una apariencia necesariamente agradable para el usuario por defecto. Con esa finalidad surge CSS (*"Cascading Style Sheets"*, hojas de estilo en cascada, en inglés).

En los inicios de la Web, el estilo del documento (colores e imágenes de fondo, colores y tamaños de letra) estaba incrustado mediante atributos en las etiquetas. Tras el crecimiento del número de páginas en HTML, se vio la necesidad de mantenimiento de las mismas y que, para cambiar un simple detalle, había que retocar en un sin fin de apartados. Es entonces cuando se decide que es interesante poder separar la estructura y el estilo del documento. De esta manera, se puede cambiar el estilo simplemente sustituyendo un archivo.

2.3.2.1 Posibilidades que ofrece CSS

Mediante CSS es posible alterar prácticamente cualquier detalle de la apariencia del contenido de nuestra Web: color, tamaño, separación a los lados con otros elementos, alineación del texto, imágenes de fondo, modificar los estilos propios impuestos por el navegador que se utilice y todo ello mediante un lenguaje de estilo que hace uso de *selectores*, descritos posteriormente, y que ha ido evolucionando con el tiempo hasta llegar a la definición actual, que a la fecha de escritura de este documento es "CSS3".

CSS3 aglutina todas las posibilidades que ofrecían las versiones anteriores, añadiendo otras como: la posibilidad de generar transiciones y animaciones en 3D, degradados de color, bordes redondeados, rotaciones de textos, fondos múltiples simultáneos para un elemento (muy útil en los videojuegos y animaciones), y muchos más[URL8]. Todas estas novedades son llevadas a cabo por el hardware, lo que es mucho más veloz[URL9].

2.3.2.2 Formas de usar CSS:

Hay varias formas de utilizar los estilos CSS en los documentos HTML, como podemos ver en Wikipedia [10][URL10]:

Método "inline"

Este método consiste en incluir el código CSS dentro de las propias etiquetas HTML mediante el atributo "style". Esta opción hoy en día no es la más correcta, pues este código puede resultar difícil de mantener en el tiempo, pero inicialmente, cuando las páginas HTML eran elementos más estáticos y no existía tanto dinamismo en la web era el método más extendido. Se puede ver con un ejemplo:

```
<div style="border-top: 1px solid black; font-weight:bold">
```

```
Esto es un div  
</div>
```

Aunque, como se ha mencionado, no es una práctica muy acertada en la actualidad, sigue utilizándose en los correos electrónicos cuando se envía contenido HTML, ya que los clientes de correo electrónico pueden no soportar otro sistema de inclusión de hojas de estilo.

Mediante una hoja de estilo interna

Esta técnica consiste en crear una hoja de estilo pero incluirla dentro de etiquetas “<style>” en el contenido HTML, generalmente en la cabecera (etiqueta “<head>”). Se suele utilizar cuando es un estilo muy específico o por ejemplo en respuestas AJAX que traen sus propios estilos CSS de los elementos que devuelven. Aunque esta técnica permite reutilizar código CSS en el documento HTML (utilizando identificadores y clases), sigue siendo problemática para reutilizar estilos en varios documentos HTML o páginas diferentes. Su ventaja es que permite un maquetado HTML más limpio y fácil de mantener, separando estructura y presentación/estilos. Lo siguiente es un ejemplo:

```
<style>  
#myDiv{  
  border-top: 1px solid black;  
}  
.negrita{  
  font-weight: bold;  
}  
</style>  
  
<!-- más código -->  
<div id="myDiv" class="negrita">  
Esto es un div  
</div>
```

Mediante una hoja de estilo externa

Esta técnica podría verse como el siguiente paso, que es extraer los estilos CSS a un fichero externo con extensión “*.css”, fuera del documento HTML y referenciarlo mediante la etiqueta “link”. Con esto se consigue compartir estilos entre todas aquellas páginas que referencien este fichero. Para ello, como se muestra en el ejemplo, se ha de proporcionar la ruta donde se encuentra alojado dicho fichero:

```
<link rel="stylesheet" type="text/css" href="estilos.css"  
media="screen" />
```

2.3.2.3 Selección de elementos mediante CSS

Cuando se desea aplicar estilos CSS utilizando hojas de estilo (ya sean internas o externas al documento) se debe seleccionar a qué elementos se le aplicarán dichos estilos, ya que la definición se encuentra fuera de la etiqueta HTML cuya apariencia se quiere modificar. Esto es lo que se consigue a través de los *selectores* CSS. De forma resumida, mediante los selectores se puede indicar si se quieren aplicar unos estilos a todas las etiquetas HTML de un tipo concreto (por ejemplo `<div>`, `<p>...`), a etiquetas concretas anidadas a su vez dentro de otro tipo concreto, a aquellos elementos que tengan un cierto identificador ("id"), o a elementos con una clase o combinación de clases.

En la tabla 5 [CSS,2] se puede ver una relación básica de selectores aplicables:

Selector	Descripción
*	Selector universal, son todos los elementos del CSS
E	E representa cualquier elemento del tipo E (span, p, ...)
E F	Todos los elementos F que sean descendentes de E
E > F	Todos los elementos F que sean hijos de E
E:first-child	De esta forma podemos seleccionar el primer elemento de tipo E
E:link , E:visited	Selecciona los elementos E que sean un enlace y no hayan sido visitados (:link) y los sí visitado (:visited)
E:active , E:hover , E:focus	Selecciona los elementos de tipo E, en sus correspondientes <u>acciones</u> .
E:lang(c)	Cogemos los elementos del tipo E que estén en el idioma (humano) especificado en (c).
E + F	Se trata de cualquier elemento F inmediatamente después del elemento del tipo E
E[foo]	Elementos del tipo E con el atributo foo
E[foo="ejemplo"]	Elementos del tipo E con el atributo foo igual a "ejemplo"
E[foo~="ejemplo"]	Elementos del tipo E con el atributo foo contenga "ejemplo". Se pueden añadir varias palabras separadas por espacios.
E[lang]="es"]	Similar al anterior, pero se referirá a todos los elemento E tal que su atributo lang comience por "es". Por ejemplo: "es_ES", "es_CA",...
E[foo\$="ejemplo"]	Elementos del tipo E en el que el atributo foo termine con "ejemplo".
DIV.ejemplo	Todos los elementos DIV que sean de la clase ejemplo
E#miID	El elemento E en el que su ID sea igual miID (En este caso se puede obviar el elemento E, ya que el miID debería ser único)

Tabla 5. Selectores CSS básicos

2.3.3 Javascript

Javascript es un lenguaje de programación interpretado, que se utiliza principalmente del lado del cliente, aunque recientemente ha empezado a utilizarse también desde el lado del servidor gracias a proyectos como el de “Node.js”.

Mediante Javascript no solamente se puede agregar una lógica a la página Web, validando datos sino que se puede acceder a los distintos elementos que componen el árbol de etiquetas HTML anidadas que forma la página Web (conocido como “DOM”) y manipularlos, añadiendo nuevos elementos en cualquier punto donde se necesite, eliminando otros y/o modificando los que sea oportuno. Gracias a ello es posible crear animaciones o una interacción dinámica con el usuario, alterando la página según las acciones que vaya tomando. Se permite, de igual manera, alterar los estilos CSS de la página o validar si lo que ha introducido el usuario en un formulario tiene el formato correcto esperado o necesitado por nuestra aplicación.

Hoy en día, en el ámbito Web es frecuente la utilización de librerías ampliamente aceptadas por los desarrolladores que simplifican mucho la programación y el acceso a los elementos. Una de las librerías Javascript con más aceptación y comunidad hoy en día es “jQuery”, que además cuenta con una versión o extensión para móviles, que no sólo facilita la interacción con el DOM sino que también aplica estilos CSS adaptados a móvil al documento HTML.

2.3.3.1 jQuery

jQuery es una librería Javascript de código libre y abierto originada en 2006, escrita por John Resig, que permite acceder muy fácilmente a los distintos elementos del DOM mediante selectores. Estos selectores son los mismos que utiliza CSS, por lo que además facilita el acceso a la programación básica en JavaScript a los diseñadores Web.

jQuery, al igual que otras bibliotecas de código, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio[URL13].

2.3.3.2 JQuery Mobile

jQuery Mobile está construido sobre la librería jQuery, y en su página Web[JqM, 1] lo definen como (traducido del inglés):

“Un sistema de interfaz de usuario, unificado y basado en HTML5, para todas las plataformas móviles más populares y construido en la sólida base de jQuery y jQuery UI. Su código ligero está desarrollado bajo mejoras progresivas y tiene un

diseño flexible y al que se pueden agregar plantillas de colores (conocidas como “themes” en inglés).”

Como indica en su Web, actualmente soporta las siguientes plataformas móviles:

1. iOS
2. Android
3. Blackberry
4. Bada
5. WindowsPhone
6. Palm WebOS
7. Symbian
8. MeeGo

jQuery Mobile, mediante el seguimiento de ciertas pautas en el etiquetado HTML y valiéndose de los atributos “data-xxx” de HTML5 (*conocidos como “custom data attributes”*) genera una salida por pantalla con unos estilos CSS optimizados para dispositivos móviles[URL15],

En la figura 6 podemos ver una captura de pantalla de una aplicación Web móvil[URL16] que utiliza jQuery Mobile, donde podemos ver los estilos aplicados por la librería para un elemento de lista no ordenada (“”), entre otros.

2.3.4 Compatibilidad de HTML5 entre los diferentes navegadores de terminales móviles

A la hora de desarrollar una aplicación Web, no solamente se ha de tener en cuenta lo que el desarrollador es capaz de hacer en la aplicación de manera técnica y visual para generar una buena experiencia de usuario, sino que también es preciso tener en cuenta lo que los navegadores Web soportan y cómo muestran la información al usuario. Tal vez el punto fuerte de la aplicación se base en el uso de “Websockets” para reducir el tráfico de datos y evitar el “polling”, o tal vez se base en un uso intensivo de algunas características CSS3 que hacen la aplicación muy atractiva. Antes de nada, se ha de tener en cuenta la compatibilidad entre los distintos navegadores Web y adaptar el desarrollo de manera que la aplicación se ejecute correctamente en todos ellos o al menos en los que se consideren más importantes.

En terminales de escritorio, las opciones más comunes son mucho más reducidas: según Universia, en el primer trimestre de 2011 la inmensa mayoría accede desde Internet Explorer, Mozilla Firefox, Google Chrome, Opera o Safari[URL17]. En cambio, en los terminales móviles y tabletas las opciones pueden ser más amplias. No obstante, los dispositivos móviles vienen con un navegador Web propio y nativo. Mientras que usuarios más avanzados sustituirán por navegadores más populares que satisfagan mejor sus necesidades como, por ejemplo, Chrome para Android, los usuarios estándar tenderán a utilizar el navegador que viene por defecto con el terminal.

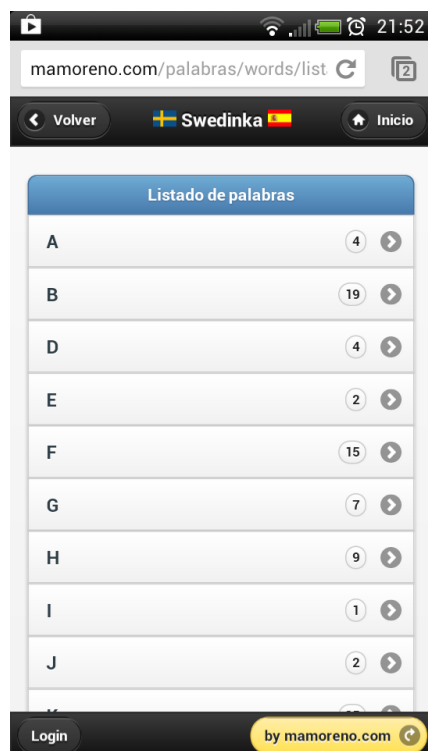


Figura 6. Captura de pantalla de una aplicación Web móvil que utiliza jQuery Mobile

Por lo tanto, el desarrollador se enfrenta al problema de compatibilidad entre navegadores y las funciones que ofrecen. Para hacer que la aplicación se ejecute en el mayor número de terminales posibles, se deben conocer bien sus características con el fin de utilizar las más aceptadas en el momento de desarrollo, teniendo en cuenta la popularidad de uso de los navegadores. Para conocer estos datos se puede acceder a recursos como <http://caniuse.com> que ayudan a identificar qué navegadores (escritorio o móviles/tabletas) implementan o no una característica HTML5. Para el caso específico de navegadores para móviles y tabletas, se puede acceder a <http://mobilehtml5.org/> donde se mantiene una tabla actualizada con esta información.

Search:

1 result found

Index Tables

Compatibility tables Browser comparison

Show options

■ = Supported
 ■ = Not supported
 ■ = Partially supported
 ■ = Support unknown

Web Sockets - Working Draft

Bidirectional communication technology for web apps

Usage stats: Global

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
7.0								2.1	
8.0								2.2	
9.0								2.3	
Current	7.0	16.0				3.2		3.0	
Near future	8.0	17.0	23.0	5.1		4.0-4.1		4.0	
Farther future	9.0	18.0	24.0	6.0	12.1	4.2-4.3		4.1	
		19.0	25.0		12.5	5.0-5.1		4.1	
		20.0	26.0					4.2	7.0
							5.0-7.0		10.0

Figura 7. Resultados de “caniuse.com” para la funcionalidad HTML5 Websockets

Feature	Safari on iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser		Nokia Browser		Internet Explorer		Opera Mobile	Opera mini	Firefox	webOS Browser	
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones	Tablet	MeeGo + N9	Symbian	Windows Phone 7.5	Windows 8	Android & Symbian	Java iOS Android	Android, MeeGo	HP Phones	HP TouchPad
Versions tested	3.2 to 6.1	1.5 to 4.2	19 to 29b	1.0 to 2.0	5.0 to 7.1	BB10	1.0 to 2.1	1.2	*3 to Belle FP2	9	10 (metro)	11 to 12.1	5 to 7	6 to 18	1.4 to 2.0
Application Cache W3C API Offline package installation.	✓	✓ 2.1+	✓	✓	✓ 6.0+	✓	✓	✓	✓	✓	✓		✓	✓	✓
Web storage W3C API Persistent and session storage.	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓	✓	✓	✓	✓	✓		✓	✓	✓
Web SQL storage W3C API (no active) Persistent SQLite storage.	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓	✓	✓	✓		✓			✓	✓
Geolocation W3C API Geolocation & tracking using GPS, cell or Wi-Fi.	✓	✓ 2.0+	✓	✓ 2.0+	✓ 6.0+	✓	✓	✓	✓	✓	✓		✓	✓	✓
Multimedia W3C API Video & Audio Players	✓	✓ 2.3+	✓	✓	✓ 7.0+	✓	✓	✓	✓	✓	✓		✓	✓	✓
Server-Sent Events W3C API EventSource pattern to maintain the connection to the server open	✓ 4.1+		✓	✓ 2.0+		✓ 2.0+	✓				✓		✓		
Web Sockets W3C API Newbidirectional protocol over HTTP	✓ 4.2+		✓	✓ 2.0+	✓ 6.1+	✓		✓ Belle FP2+		✓	✓ 12.1+		✓ 7+		✓ 3.0.5+

Figura 8. Ejemplo de la tabla con los datos que nos ofrece “mobilehtml5.org”

2.3.4.1 *Ventajas y desventajas de solucionar el problema propuesto mediante una aplicación Web*

Ventajas

- Se ha de desarrollar una única aplicación
- Accesible desde cualquier dispositivo con acceso a Internet
- Sin necesidad de descargas previas
- Sin intermediarios ni censuras de aplicación
- El usuario siempre accede a la versión más actualizada que queramos entregarle.

Desventajas

- Los usuarios que acceden a la Web pueden no tener el navegador o terminal apropiado, por lo que la experiencia puede ser mala. Para evitar esto, hay que invertir mucho tiempo en desarrollo y pruebas, lo que aumenta los costes.
- Es frecuente que los navegadores no guarden el estado de las páginas, lo que fuerza a que se recargue la Web accedida cada vez que se abre la aplicación del navegador. Esto puede ser muy molesto para el uso al que está destinada nuestra aplicación, ya que puede ser que tengamos documentos o imágenes ya cargados y que al recargarse nos consuma tiempo y datos por duplicado, que además puede repercutir en costes al usuario.
- Mayor consumo de datos: Cada vez que accedemos a la Web, todo lo que no esté en la caché, se vuelve a descargar una y otra vez en cada petición. Además, a la fecha de escritura de este documento (febrero de 2013) no todos los navegadores Web para móvil soportan websockets[URL18], por lo que las actualizaciones de los contenidos e interacciones se deberían hacer mediante "*polling*" (preguntar cada cierto tiempo al servidor si tiene nueva información para la aplicación), lo que contribuye a un mayor consumo de datos y también a una mayor saturación del servidor.
- Necesarios conocimientos en muchas áreas para el desarrollo: lenguajes del cliente, de servidor, de maquetado, de estilo, de intercambio de datos, etc.

2.4 Desarrollo de una solución mediante una aplicación nativa en una de las plataformas móviles más populares.

En los últimos 5 años ha habido un desarrollo y penetración de los dispositivos móviles en la sociedad moderna prácticamente imparable, llegando al punto en que en muchos países (entre ellos España) ya hay más dispositivos móviles que población. En el caso de España se estiman 115,3 líneas por cada 100 habitantes según un artículo publicado el 18 de enero de 2013[URL19]. Además, las velocidades de conexión cada vez son mayores y se ofrecen a menores precios, resultando más asequibles y haciendo que un gran porcentaje de la población pueda acceder a Internet regularmente con su dispositivo, en cualquier lugar y en cualquier momento. En la siguiente imagen podemos observar el crecimiento en penetración de teléfonos inteligentes dentro de la población estadounidense entre octubre de 2010 y febrero de 2012:

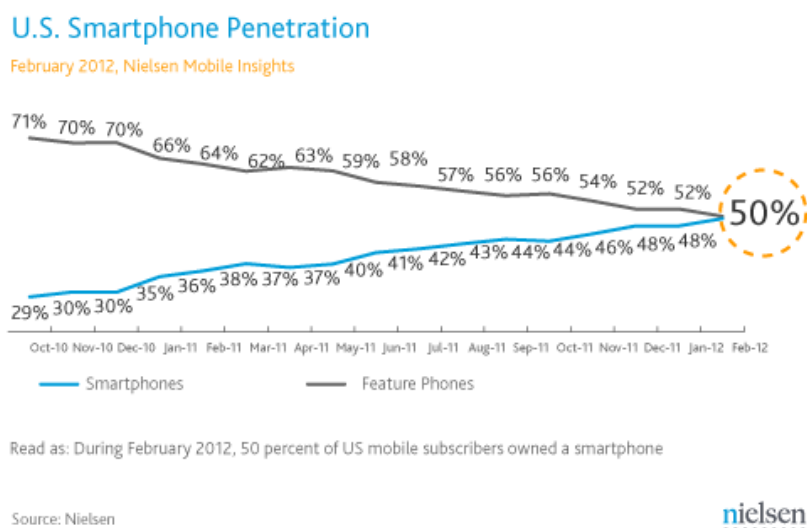


Figura 9. Penetración de Smartphones en EE.UU[URL20]

En cuanto al tipo de plataforma que eligen los usuarios, se ha de remarcar que el 48,5% de los estadounidenses utilizan el sistema de Google, seguidos por iOS con un 32% y en tercer lugar RIM Blackberry[URL21], como nos muestran desde la firma estadounidense Nielsen[URL22] de lo cual podemos extraer que el mercado se centra, en un 80%, en las plataformas Android e iOS. Además, llama la atención la escasa penetración de Windows Phone, con un 1,7%, lo que si bien puede deberse a su reciente aparición y cambio de estrategia de Microsoft en su objetivo comercial. Podemos ver una gráfica en la figura 10.

Ahora se van a describir muy brevemente las plataformas Windows Phone e iOS, ya que son, junto a Android, las principales opciones a considerar en el desarrollo para “Smartphones” hoy en día. El primero porque se está reintroduciendo en el mercado en estos momentos y el segundo porque es el otro más popular junto al sistema de Google.

2.4.1 Windows Phone

2.4.1.1 Introducción

Windows Phone es un sistema operativo móvil desarrollado por Microsoft. Se enfoca principalmente en el mercado de consumo generalista en lugar del mercado empresarial como hacía su predecesor[URL23]. Además, las aplicaciones de Windows Phone no son compatibles con Windows Mobile, por lo que han de ser reescritas para esta plataforma si se les quiere dar continuidad.

El 29 de octubre de 2012 se lanzó al mercado Windows Phone 8 destinado únicamente para nuevos dispositivos, debido a un cambio completo en el kernel que lo hace incompatible con dispositivos basados en la versión anterior. Esta versión incluye nuevas funciones que de acuerdo a Microsoft lo harán competitivo con sistemas operativos como iOS de Apple o Android de Google.

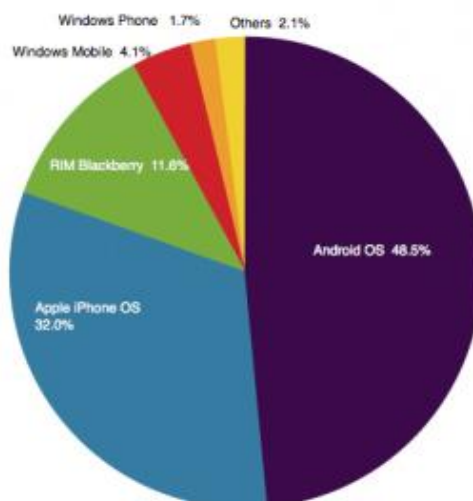


Figura 10. Distribución de SO Móviles en EE.UU[URL22]

Windows Phone tiene una nueva interfaz que se basa en un sistema mosaico personalizable y dinámico de pequeños cuadrados que pueden ser pulsados (llamados "Live Tiles")[URL24] y que son enlaces a las funcionalidades y aplicaciones instaladas en el dispositivo. La información del mosaico se va actualizando constantemente, mostrando los estados en que se encuentran las aplicaciones, por ejemplo nuevos correos, mensajes en las redes sociales, etc. Como navegador Web por defecto utiliza "Internet Explorer 10" y el sistema operativo centraliza las búsquedas del usuario en el motor "Bing", desarrollado por Microsoft. Según el CEO de Nokia, su mercado de aplicaciones (Windows Phone Store) cuenta a la fecha de este escrito con más de 130.000 aplicaciones[URL25].

2.4.1.2 Arquitectura

La arquitectura de Windows Phone está basada en diferentes capas, de manera que las aplicaciones desarrolladas para la plataforma se apoyan sobre las capas inferiores. En la Figura 11[URL29] se puede observar su arquitectura:

Aplicaciones: Esta es la capa donde se ejecutan las aplicaciones y sobre la que van a ejecutar las aplicaciones que se desarrollen y que tienen accesibles los servicios ofrecidos por las capas inferiores para, por ejemplo, ejecutar la interfaz de usuario y de manera más indirecta utilizar el acelerómetro o la radio.

En el siguiente nivel conviven 3 funcionalidades diferentes: Modelo de la aplicación (App Model), Modelo de la Interfaz de Usuario (UI Model), y la integración con servicios en la nube (Cloud Integration).

Modelo de la aplicación: Aquí es donde se gestiona la aplicación, sus actualizaciones o donde se comparten datos con otros procesos/aplicaciones.

Modelo de la Interfaz de Usuario: Ofrece la posibilidad de crear interfaces de usuario ricas y complejas, basadas en eventos táctiles.

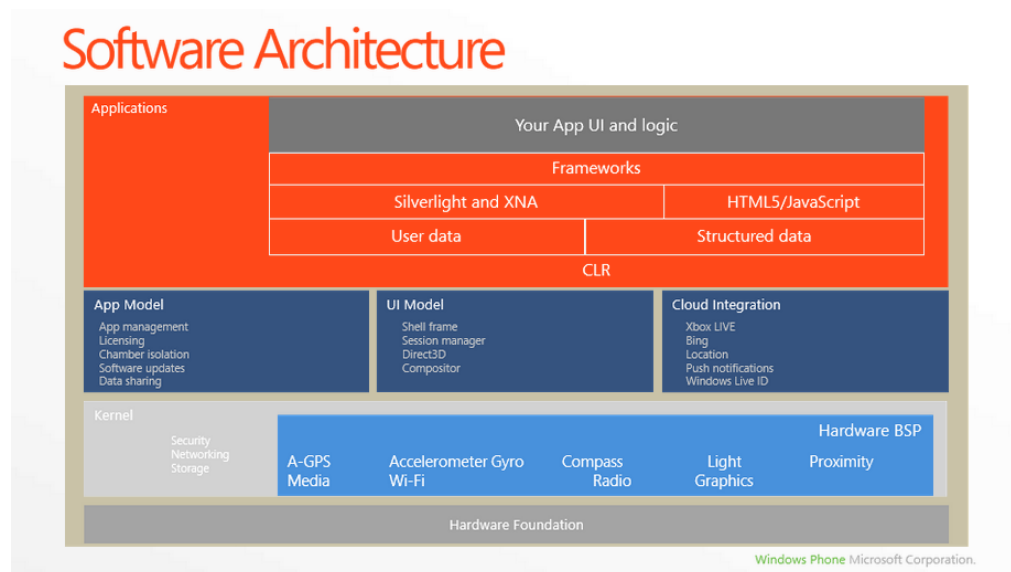


Figura 11. Arquitectura Windows Phone

Integración con la nube: Mediante esta capa podemos interactuar con diferentes servicios en la nube como: Xbox Live, Bing, Location, Push Notifications. Como ya se ha comentado anteriormente, el sistema basa sus búsquedas en la red en el sistema Bing. Esta es la capa que hace posible esta integración.

Kernel: Mediante esta capa tendremos acceso a los servicios que ofrece el hardware del dispositivo, como el GPS, la brújula, el Wi-Fi o el acelerómetro.

2.4.1.3 Sistema de venta y distribución de aplicaciones

Para la venta y distribución de aplicaciones para el sistema Windows Phone se utiliza la “Windows Phone Store”(<http://www.windowsphone.com/es-es/store>). En la Windows Phone Store el usuario puede buscar contenidos de todo tipo, entre ellos aplicaciones gratuitas y de pago, que se pueden pagar mediante una cuenta Xbox Live.

A la fecha de la redacción de este proyecto, hay más de 130.000 aplicaciones subidas a la tienda virtual.

2.4.2 iOS

2.4.2.1 Introducción

iOS es un sistema operativo desarrollado en 2007 por Apple Inc. para ejecutarse sobre dispositivos iPhone pero que poco a poco fue extendiéndose a otros dispositivos de la misma marca: iPod Touch, iPad y Apple TV. Actualmente es el segundo de los sistemas operativos más utilizados en dispositivos móviles en Estados Unidos, seguido de RIM Blackberry y adelantado por Android de Google.[URL26]

Aunque en cuanto a nivel de ingresos es la primera plataforma del mercado, es también la plataforma con mayores costes para el desarrollo sin llegar a ser prohibitivos, ya que para poder desarrollar comercialmente aplicaciones para iOS (bajo licencia "iOS Developer Program"[URL27]) hay que pagar una licencia de 99 dólares americanos anuales y además perciben el 30% del precio de la aplicación, mientras que el desarrollador solamente percibe en torno al 50-55% del coste de la aplicación, siendo el resto impuestos a cargo del desarrollador. No obstante, un desarrollador iOS puede percibir 4 veces más beneficios que un desarrollador Android[URL28].

Además, para desarrollar para iOS se necesita tener un ordenador de Apple con las actualizaciones al día, asumiendo los costes asociados a ello.

El entorno iOS se caracteriza principalmente por:

Hardware homogéneo: Desde que aparecieran los dispositivos de Apple por primera vez, el ritmo de aparición de nuevos terminales es de aproximadamente uno al año, lo que hace que la variedad de hardware sea muy homogénea y los desarrolladores puedan concentrar esfuerzos en desarrollar la aplicación pensando en el software del teléfono y no en si el terminal tendrá o no ciertas características, pudiéndolas dar por presentes.

Estabilidad en el software: Las actualizaciones de software son frecuentes y están aseguradas para todos los dispositivos, lo que ayuda a que la gran mayoría de los usuarios tengan la versión más reciente del software. Al desarrollador, esto le proporciona estabilidad y la confianza de que su aplicación seguirá funcionando en el futuro.

Entorno de desarrollo: El entorno de desarrollo para iOS es considerado uno de los más robustos y completos aunque solamente ejecuta sobre ordenadores de Apple. Como lenguaje de desarrollo utiliza "Objective-C", que es un super conjunto del lenguaje C.

Plataforma de venta: Las aplicaciones pueden venderse y/o distribuirse a los futuros usuarios a través de la aplicación de Apple Inc. llamada "App Store" y que acompaña al sistema operativo de todos los dispositivos móviles de la marca.

2.4.2.2 Arquitectura

El software que se desarrolla para iOS muy raramente se comunica con el hardware de manera directa. En su lugar, la aplicación se comunica a través de un sistema de interfaces definidas para este fin.

La implementación de iOS puede verse por tanto como una arquitectura por capas[URL30], tal y como se puede ver en la figura 12 [URL31], cada una con un cometido:

Cocoa Touch: Esta capa es la que lleva a cabo las acciones de alto nivel y con las que el usuario realmente tiene contacto. En este nivel se tiene lo necesario para manipular documentos, detectar los gestos que el usuario realiza en la pantalla táctil, dibujar la interfaz con la que podrá interactuar, etc.

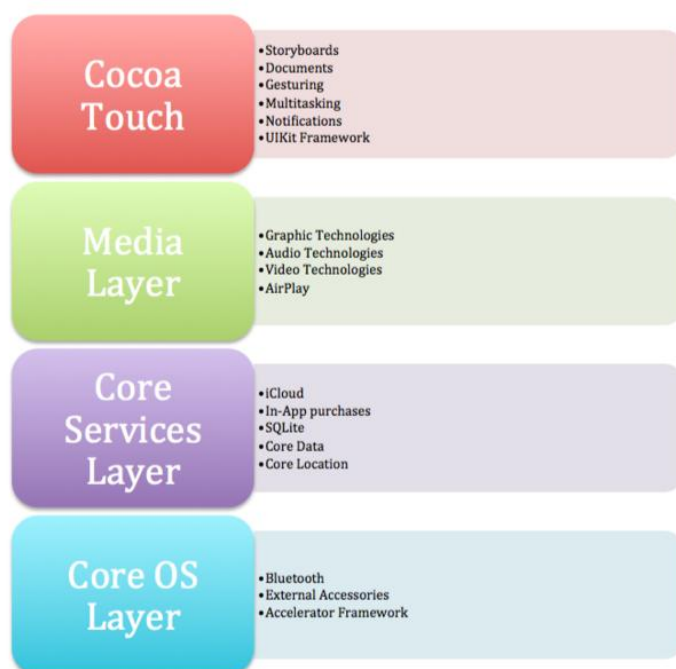


Figura 12. Arquitectura por capas de iOS

Media Layer: En la capa “Media” se tienen las herramientas necesarias para todo el contenido multimedia de la aplicación, ya sea de tipo audio, video, imágenes...

Core Services Layer: Este nivel es el que contiene todos los servicios que ofrece el núcleo del sistema operativo, como el almacenamiento de contenidos en una base de datos SQLite², en ficheros XML o incluso en la nube mediante iCloud³.

Core Layer: Es la capa de más bajo nivel, que se comunica con el Hardware y sobre la que la mayoría de las otras tecnologías se ha construido. Se encarga de

² Este sistema de bases de datos se define en un apartado posterior

³ iCloud es el sistema de Apple para almacenaje en la nube

las transmisiones por Bluetooth, y de todo lo relativo a los servicios del sistema operativo de bajo nivel, como son los sockets, el sistema de ficheros o el manejo de la concurrencia y la gestión de memoria.

2.4.2.3 Sistema de venta y distribución de aplicaciones

La tienda virtual App Store es un servicio ofrecido por Apple que permite a los usuarios buscar y descargar aplicaciones para iPod Touch, iPad, Mac OS X Snow Leopard y superiores. Las aplicaciones que aloja pueden venderse o pueden ser gratuitas, pero en ambos casos, y como se mencionó anteriormente, se ha de poseer una licencia para publicar aplicaciones y que estas pasen los filtros de calidad de Apple. En el caso de que las aplicaciones no sean gratuitas, Apple se queda con un porcentaje del precio de venta.

2.4.3 Android

2.4.3.1 Breve Historia de Android

Android, Inc. fue fundada por Andy Rubin en octubre de 2003 en California, fue comprada por Google por un valor de 50 millones de dólares en agosto de 2005[URL32].

Durante 2006 lanzan su primer prototipo y en mayo de 2007 el primer dispositivo Android: Google Sooner (htc-2065.0.8.0.0) el cual tenía solamente 64MB de RAM y teclado físico[URL33].

Al poco tiempo de haberse lanzado el primer terminal con Android como sistema operativo se crea la "Open Handset Alliance" (noviembre de 2007), que es una alianza comercial de 84 compañías que se dedica a desarrollar estándares abiertos para dispositivos móviles. Algunos de sus miembros son Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia y Wind River Systems, que comparten la meta de dar a los consumidores una experiencia de usuario mucho mejor cada día[URL34].

No obstante, tuvo que pasar casi otro año, hasta septiembre de 2008, para que se lance el primer dispositivo Android a producción, producido por T-Mobile y llamado "G1", con sistema Android 1.0, lanzándose a la vez el primer SDK de desarrollo para Android y al mes siguiente, octubre de 2008, Google y la Open Handset Alliance liberan el código de Android. Esto permite a los desarrolladores implementar ROMs (versiones modificadas del Sistema Operativo Android) propias.

A partir de ese momento y con bastante frecuencia se van lanzando nuevas versiones del sistema operativo con nuevas funcionalidades, pero no fue hasta principios de 2011 que Android le comió cuota de mercado a iOS en Estados Unidos y lanzó su versión 3.0 (Honeycomb) que es específica para tabletas, renovando el aspecto de la interfaz.

A la fecha de este escrito, Android se encuentra ya en su versión 4.1 (Jelly Bean), se han activado mundialmente más de 400 millones de dispositivos con alguna versión del sistema operativo Android y su aplicación de difusión de contenidos (Android Market y ahora Google Play Store, accesible como aplicación preinstalada en Android y en <https://play.google.com/store>) ya ha registrado más de 25 mil millones de descargas y cuenta con más de 675 mil aplicaciones.

Versiones de Android[URL35]


	Nombre	Cupcake
	Versión	1.5
	Lanzamiento	Abril de 2009
	Funciones	Teclado QWERTY virtual, widgets, captura de video, copiar&pegar, subir videos a Youtube en forma directa
	Notas	Marca el inicio de este sistema operativo

Figura 13. Logo de Android Cupcake

Tabla 6. Resumen de Android Cupcake


	Nombre	Donut
	Versión	1.6
	Lanzamiento	Septiembre de 2009
	Funciones	Navegación en Google Maps paso-a-paso, mejora en el interfaz de la cámara de foto y video, búsquedas por voz.
	Notas	Inicia de manera importante el crecimiento de la comunidad de desarrolladores.

Figura 14. Logo de Android Donut

Tabla 7. Resumen de Android Donut


	Nombre	Eclair
	Versión	2.0
	Lanzamiento	Diciembre de 2009
	Funciones	Zoom digital de la cámara de fotos y video, mejoras en Google Maps, nuevo navegador de internet con soporte de video e incluso de Flash, salvapantallas animadas.
	Notas	Comienza a demostrar la robustez y estabilidad como sistema operativo para dispositivos móviles.

Figura 15. Logo de Android Eclair

Tabla 8. Resumen de Android Eclair


	Nombre	Froyo (Frozen Yoghourt)
	Versión	2.2
	Lanzamiento	Mayo 2010
	Funciones	Mejora de memoria, mucho más veloz que las versiones anteriores, hotspot WiFi incluido, soporte de correo electrónico Microsoft Exchange y video llamada. Soporta Flash.
	Notas	Empieza a encontrar su sitio entre la mayoría de los dispositivos móviles a nivel mundial. Excelente aceptación por parte de los usuarios.

Figura 16. Logo de Android FroYo

Tabla 9. Resumen de Android Froyo



Figura 17. Logo de Android Gingerbread

Nombre	Gingerbread
Versión	2.3
Lanzamiento	Diciembre de 2010
Funciones	Mejora del soporte de video online, mejora de la gestión de la energía (y consumo de la batería), mejora del teclado virtual, soporte para pagos mediante NFC.
Notas	Abre más posibilidades para usuarios totalmente conectados.

Tabla 10. Resumen de Android Gingerbread



Figura 18. Logo de Android Honeycomb

Nombre	Honeycomb
Versión	3.0/3.4
Lanzamiento	Mayo de 2011
Funciones	Sistema operativo optimizado para tabletas, posee plenas funcionalidades multitarea para cambiar de aplicaciones en uso dejando las demás en espera visibles en una columna, soporte Flash y DivX, aparte de los navegadores privados tradicionales integra Dolphin, home page personalizable y "widgets".
Notas	Versión optimizada para tabletas, mantiene en "Smartphone" la versión 2.3

Tabla 11. Resumen de Android Honeycomb



Figura 19. Logo de Android Ice cream sándwich

Nombre	Ice cream sandwich
Versión	4.0
Lanzamiento	Octubre de 2011
Funciones	Diseñado para todas las plataformas (smartphones, tabletas y netbooks de todas las medidas), pantalla principal con imágenes 3D, barras de estado y "widgets" redimensionables, reconocimiento facial, mejora de reconocimiento de voz; soporte USB para teclados y controles para PS3.
Notas	Novedades y mejoras en la navegación e interfaz, con el objetivo de dinamizar y facilitar la experiencia del usuario.

Tabla 12. Resumen de Android Ice Cream Sandwich


	Nombre	Jelly Bean
	Versión	4.1 /4.2
	Lanzamiento	Junio de 2012
	Funciones	Optimización de las transiciones en la interfaz (proyecto Butter ⁴), mejor pantalla de inicio con “ <i>widgets</i> ” e iconos cuyos tamaños se ajustan automáticamente, mejor vista previa de las capturas fotográficas, mejora en la predicción de palabras del teclado y soporte offline para introducción de texto por voz, dieciocho nuevos idiomas y mejor accesibilidad, con soporte Braille, Android Beam (tecnología NFC) para compartir archivos entre dispositivos con NFC, mejora de la barra de notificaciones desde donde se puede devolver una llamada, leer el comienzo de un email, interactuar en redes sociales, etc., mejora en el reconocimiento de voz de Google, incluye Google Now ⁵ .
	Notas	http://www.taringa.net/posts/noticias/15103488/Android-4-1-Jelly-Bean-Nuevas- Caracteristicas.html

Figura 20. Logo de Android Jelly Bean

Tabla 13. Resumen de Android Jelly Bean

Además, mientras se escribe este documento se habla de la versión “Key Lime Pie”, versión 5.0 y que incluirá nuevas funcionalidades como chat con video, múltiple selección de contactos, teclado con escritura de trazos... etc.[URL36]

⁴ En el Proyecto “*Butter*” se cristaliza el esfuerzo de Google por mejorar la experiencia de usuario en Android, haciendo más suaves las transiciones de la interfaz. Android escalará la velocidad de las CPU tan pronto como el usuario toque la pantalla. La versión 4.1 también predecirá qué tocará el usuario a continuación, manteniendo el sistema operativo siempre un paso por delante.

⁵ Google Now hace uso del historial de búsquedas y navegación del usuario, de su actual posición y de sus próximos eventos anotados en el calendario, para avisarle de cuándo debe marcharse para acudir a su próxima cita, o si su vuelo se ha retrasado, y lo tiene en cuenta para las tareas anotadas en el calendario. Google Now también muestra información de los resultados del equipo deportivo favorito del usuario y le recomendará restaurantes cercanos.

2.4.3.2 Cuota de mercado de las diferentes versiones Android

Según recoge la propia web para desarrolladores de Android[URL37] y según datos recogidos a fecha de 2 de Abril de 2013, la cuota de mercado de las diferentes versiones Android es la siguiente:

Versión de Android	Nivel de API	% de implantación
4.x.x - Jelly Bean	16-17	25,0%
4.0.x - Ice Cream Sandwich	14-15	29,3%
3.x - HoneyComb	12-13	0,2%
2.3.x - Gingerbread	9-10	39,8%
2.2 - Froyo	8	4,0%
2.1 - Eclair	7	1,7%
1.6 - Donut	4	0,1%
1.5 - Cupcake	3	Menor al 0,1%

Tabla 14. Cuota de mercado de las diferentes versiones Android (abril '13)

Lo que se puede representar en el siguiente gráfico (obtenido de la misma fuente que los datos)

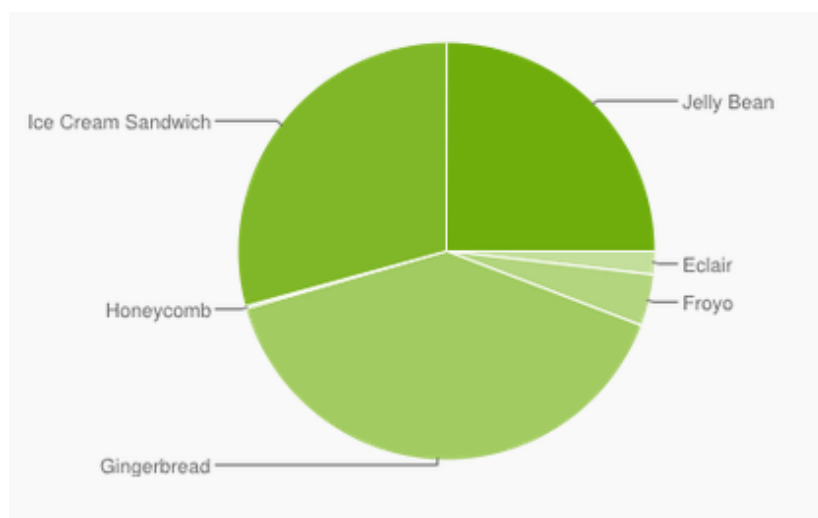


Figura 21. Distribución de la cuota de mercado entre las diferentes versiones

Lo que indica que, desarrollando a partir de Gingerbread, se puede dirigir la aplicación a más del 94% de los dispositivos Android existentes.

2.4.3.3 Arquitectura Android

La plataforma Android contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario[URL38]. Estas capas se organizan o apilan como se observa en la figura 22[URL39].



Figura 22. Pila software de Android

Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores. Internamente, cada una de estas capas está compuesta de los componentes que se observan en la figura 23[URL40].

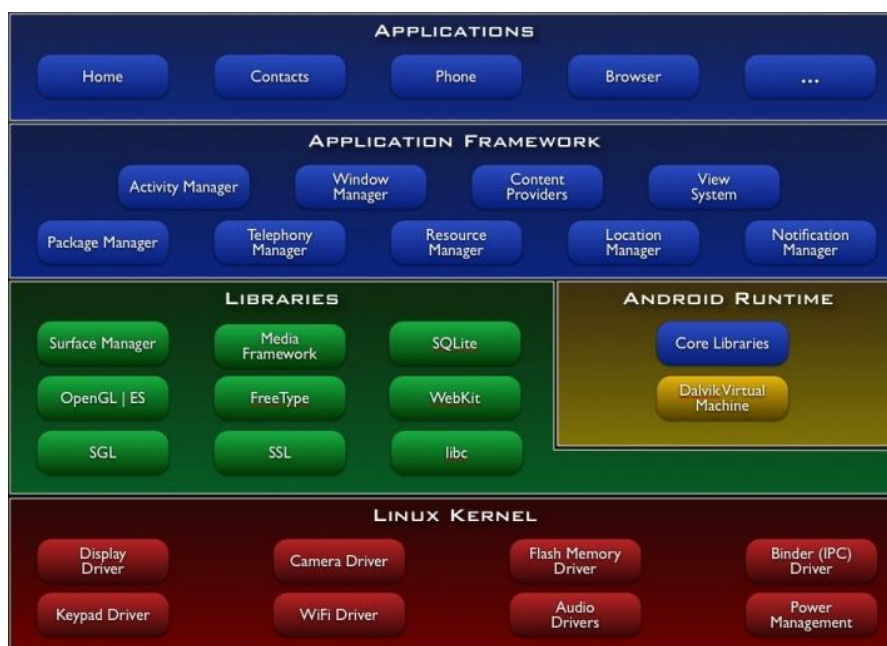


Figura 23. Arquitectura Android

Aplicaciones: Este nivel contiene tanto las incluidas por defecto de Android (teléfono, navegador, contactos...) como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API's y librerías de los niveles anteriores.

Framework de Aplicaciones: Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "Framework" representado por este nivel. Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **Activity Manager:** Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.
- **Window Manager:** Gestiona las ventanas de las aplicaciones. Utiliza la librería Surface Manager.
- **Telephone Manager:** Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.).
- **Content Provider:** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Gracias a esta API la información de contactos, agenda, mensajes, etc. es accesible para otras aplicaciones y de esta manera se puede, por ejemplo, compartir una foto de la galería mediante twitter.
- **View System:** Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, "check-boxes", tamaño de ventanas, control de las interfaces mediante teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.
- **Location Manager:** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- **Notification Manager:** Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada Intent, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.
- **XMPP Service:** Colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

Librerías: La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android.

Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **Librería libc:** Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- **Librería Surface Manager:** Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.

- **OpenGL/S� y SGL:** Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/S� maneja gráficos en 3D y permite utilizar, **en caso de que esté disponible en el propio dispositivo móvil**, el hardware encargado de proporcionar gráficos 3D. Por otro lado, SGL proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.
- **Librería Media Libraries:** Proporciona todos los *codecs* necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- **FreeType:** Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- **Librería SSL:** Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- **Librería SQLite:** Creación y gestión de bases de datos relacionales. Se hablará de SQLite en otro apartado.
- **Librería WebKit:** Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Android Runtime: Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Éste lo constituyen las *Core Libraries*, que son librerías con multitud de clases Java y la máquina virtual Dalvik.

Núcleo Linux: Android utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluye un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro de este “*kernel*” de Linux embebido en el propio Android.

2.4.3.4 Componentes de una aplicación

Los componentes de la aplicación son los bloques esenciales sobre los que se construyen las aplicaciones Android. Cada componente es un punto diferente a través del cual el sistema puede comunicarse con una aplicación. No todos los componentes son puntos de entrada para el usuario y algunos dependen entre sí, pero cada uno existe en sí mismo y juega un rol específico.

Hay cuatro tipos diferentes de componentes de aplicación. Cada uno sirve a un propósito distinto y tiene un ciclo de vida diferente que define como se crea y destruye el componente[URL41].

Estos son los cuatro tipos de componentes:

1. **Actividades:** Una actividad representa una única pantalla de la interfaz de usuario (por ejemplo el listado de contactos de la agenda) y una aplicación Android puede tener cero o muchas actividades. Aunque las actividades funcionan colaborando entre sí, son independientes unas de las otras y por ello diferentes aplicaciones pueden lanzar cada una de estas actividades por separado (si la aplicación lo permite). Una actividad tiene un ciclo de vida muy definido, que será igual para todas las actividades. Este ciclo de vida es impuesto por el SDK de Android. Las actividades tienen cuatro posibles estados (Activa, pausada, parada y reiniciada) que interaccionan entre sí siguiendo el grafo representado por la figura 24[URL42].

Las actividades se implementan como subclases de la clase "Actividad".

2. **Servicios:** Un servicio es un componente que se ejecuta en silencio para llevar a cabo tareas de actualización de contenidos, comprobaciones, mantener la ejecución mientras la aplicación no está a la vista... etcétera. Otro componente, como una actividad, puede lanzar la ejecución de un servicio para poder interactuar con él. Un servicio se implementa como subclase de la clase "Service".

3. **Proveedores de contenidos** (Content Providers): Un proveedor de contenido maneja un conjunto compartido de datos. Puedes almacenar datos en el sistema de ficheros, en una base de datos SQLite, en la Web o en cualquier otro sistema persistente de almacenamiento que tu aplicación pueda acceder. A través del proveedor de contenidos, otras aplicaciones pueden extraer o incluso modificar los datos (siempre y cuando el proveedor lo permita). Por ejemplo, los contactos. A través del proveedor de contenidos, cualquier otra aplicación puede leerlos o modificarlos. Los proveedores de contenidos también son de utilidad para almacenar y leer información que es privada a la aplicación y no se comparte, como se puede hacer mediante ficheros en otros sistemas.

Un proveedor de contenidos se implementa como subclase de "ContentProvider" y ha de implementar una serie de métodos que permitan a las aplicaciones llevar a cabo las transacciones de escritura y lectura.

4. **Receptores de difusión** (Broadcast receivers): Un receptor de difusión es un componente que responde a los anuncios difundidos a todo el sistema por parte de otras aplicaciones. Esto puede ser, por ejemplo, el hecho de apagar la pantalla, batería baja, se ha tomado una fotografía, terminado de descargar un archivo, etc. El sistema hace la difusión del evento a las aplicaciones para que éstas actúen en consecuencia si así lo necesitan.

Aunque un receptor de difusión no tiene interfaz de usuario que mostrar, pueden crear una alerta en la barra de notificaciones para dejar saber al usuario que algo ya ha ocurrido.

Un receptor de difusión se implementa como subclase de "Broadcast Receiver" y cada difusión se envía a un objeto de tipo "Intent".

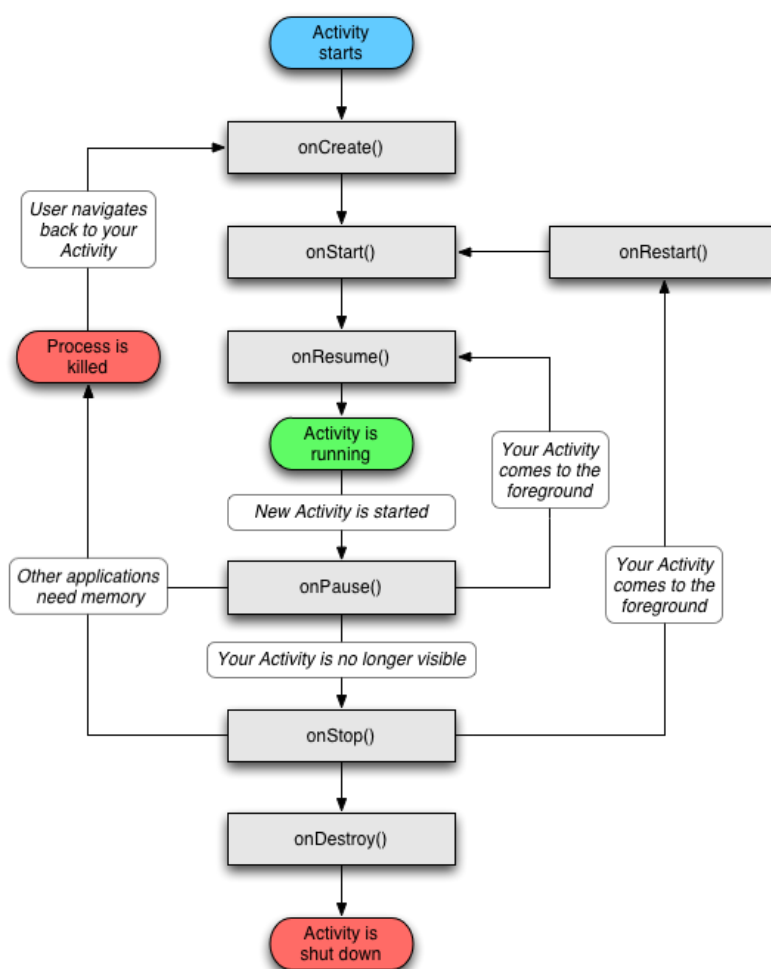


Figura 24. Ciclo de vida de una actividad en Android

2.4.3.5 Activación de componentes

A excepción de los proveedores de contenidos, los componentes anteriormente descritos se activan mediante mensajes asíncronos llamados "Intents". Los "Intents" unen componentes entre sí en tiempo de ejecución, como si fuera el mensajero que recoge una petición y la hace llegar a su destinatario (otro componente), ya sea de la misma aplicación u otra distinta.

Para las **actividades y servicios**, un "intent" define una acción a ejecutar (como ver o enviar datos, por ejemplo) y debería especificar la URI de los datos sobre los que actuar, entre otras cosas que el componente a inicializar necesite saber.

Para los **receptores de difusión**, el "intent" simplemente define el anuncio que se está difundiendo, mientras que los **proveedores de contenidos** no se activan mediante "intents", sino que en su lugar se activan cuando se les solicita por un "ContentResolver" (un intérprete de contenidos).

2.4.3.6 Android Manifest

Todas las aplicaciones Android tienen que tener un archivo XML con el **nombre exacto "AndroidManifest.xml"** en su directorio base. El manifiesto presenta información esencial sobre la aplicación al sistema operativo, una información que debe conocer antes de ejecutar cualquier trozo de código de la misma. Entre otras cosas, el manifiesto sirve para lo siguiente:

- Da nombre al paquete Java de la aplicación, lo que sirve como identificador único para la misma.
- Describe los componentes de la aplicación (actividades, servicios, receptores de difusión y proveedores de contenidos). Registra las clases que implementan cada componente y hace públicas sus capacidades.
- Determina los permisos que la aplicación tiene que tener para ejecutar (acceso a Internet, escribir en la tarjeta de memoria...).
- Enumera los permisos que otras aplicaciones deben tener para poder interactuar con la actual.
- Declara el mínimo nivel de API que la aplicación requiere para funcionar (la versión mínima concreta).
- Enumera las librerías que la aplicación debe enlazar.

2.4.3.7 Estado de los procesos en Android

Como se ha mencionado con anterioridad, cada aplicación de Android corre en su propio proceso, el cual es creado por la aplicación cuando se ejecuta y permanece hasta que la aplicación deja de trabajar o el sistema necesita memoria para otras aplicaciones. Android sitúa cada proceso en una jerarquía de "importancia" basada en estados:

- **Procesos en primer plano o activos:** Es un proceso que aloja una actividad en la pantalla y con la que el usuario está interactuando. Este tipo de procesos serán eliminados como último recurso si el sistema necesitase memoria.
- **Procesos visibles:** Es un proceso que aloja una actividad pero no está en primer plano, ha sido pausada. Esto ocurre, por ejemplo, en situaciones donde la aplicación muestra un cuadro de diálogo para interactuar con el usuario. Este tipo de procesos solo será eliminado en caso que sea necesaria la memoria para mantener a todos los procesos del primer plano corriendo.
- **Procesos de servicio:** Es un proceso que aloja un servicio que ha sido iniciado. Este tipo de procesos no son visibles y suelen ser importantes para el usuario (conexión con servidores, reproducción de música...).
- **Procesos en segundo plano:** Es un proceso que aloja una actividad que no es actualmente visible para el usuario. Normalmente la eliminación de estos procesos no suponen un gran impacto para la actividad del usuario. Es muy usual que existan numerosos procesos de este tipo en el sistema,

por lo que el sistema mantiene una lista para asegurar que el último proceso visto por el usuario sea el último en eliminarse en caso de necesitar memoria (sistema LRU-Less Recently Used).

- **Procesos vacíos:** Es un proceso que no aloja ningún componente. La razón de existir de este proceso es tener una caché disponible de la aplicación para su próxima activación. Es común que el sistema elimine este tipo de procesos con frecuencia para obtener memoria disponible.

Según esta jerarquía, Android prioriza los procesos existentes en el sistema y decide cuáles han de ser eliminados, con el fin de liberar recursos y poder lanzar la aplicación requerida.

2.4.3.8 Modelo de seguridad

La mayoría de las medidas de seguridad entre el sistema y las aplicaciones derivan de los estándares de Linux 2.6, cuyo kernel constituye el núcleo principal de Android[URL43].

Cada proceso proporciona un entorno seguro de ejecución en el que, por defecto, ninguna aplicación tiene permiso para realizar ninguna operación o comportamiento que pueda impactar negativamente en la ejecución de otras aplicaciones o del sistema mismo. Por ejemplo, acciones como leer o escribir ficheros privados del usuario no están permitidas. La única forma de poder saltar estas restricciones impuestas por Android es mediante la declaración explícita de un permiso que autorice a llevar a cabo una determinada acción habitualmente prohibida.

En Android, la seguridad abarca desde el despliegue hasta la ejecución de la aplicación. Con respecto a la implementación, las aplicaciones tienen que estar firmadas con una firma digital para poder instalarlas en el dispositivo. Dicha firma se hace a través de un certificado, cuya clave privada será la del desarrollador de dicha aplicación. Mediante la firma, la aplicación lleva adjunta su autoría.

Con respecto a la ejecución, cada aplicación se ejecuta dentro de un proceso separado. Cada uno de estos procesos tiene un ID de usuario único y permanente asignado en el momento de la instalación. Esto impone un límite al proceso porque evita que una aplicación tenga acceso directo a los datos críticos para el correcto funcionamiento del terminal.

2.4.3.9 Establecer permisos en una aplicación

Para establecer un permiso para una aplicación, es necesario declarar uno o más elementos `<uses-permission>` en el manifiesto, donde se especifica el tipo de permiso que se desea habilitar. Por ejemplo, si se quisiera permitir que una aplicación pueda monitorizar mensajes SMS entrantes, en el fichero `AndroidManifest.xml` se encontraría algo como lo que sigue:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.app.myapplication"
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
</manifest>
```

El elemento `<uses-permission>` contempla una serie de atributos que definen y matizan el alcance el permiso dado:

- **android:name:** Especificación del permiso que se pretende conceder. Debe ser un nombre de alguno de los listados en la clase `android.Manifest.Permission`, donde se especifican todos los posibles permisos que se pueden conceder a una aplicación.
- **android:label:** Una etiqueta o nombre convencional fácilmente legible para el usuario.
- **android:permissionGroup:** Permite especificar un grupo asociado al permiso. Los posibles grupos se encuentran listados en la clase `android.Manifest.permission_group`.
- **android:protectionLevel:** Determina el nivel de riesgo del permiso, y en función del mismo influye en cómo el sistema otorga o no el permiso a la aplicación. Oscila entre valores desde el 0 hasta el 3.
- **android:description:** Descripción textual del permiso.
- **android:icon:** Icono gráfico que puede ser asociado al permiso.

2.4.3.10 Almacenamiento de datos

Cuando se almacenan datos de aplicación en Android, lo más común es utilizar la base de datos que viene incluida con el sistema. Esta base de datos es SQLite.

SQLite es un sistema de bases de datos relacionales transaccional, auto contenido, que no utiliza servidor ni configuración, de código abierto y gratuito que almacena la información en un único fichero en disco por cada base de datos con toda la información de tablas, datos, disparadores, vistas, etc., y que es portable entre diferentes sistemas operativos, ya sean de 32-bit o de 64-bit.[URL44]

SQLite, no permite múltiples usuarios accediendo en modo escritura a la base de datos, debido a que para el acceso a los contenidos se bloquea todo el fichero de la base de datos. Así, esta librería está especialmente indicada cuando se requiera de una gran rapidez en las consultas y nos baste que sólo un único usuario pueda realizar modificaciones[URL45]. Todo esto la hace ideal para ser

utilizada en las plataformas móviles como sistema de almacenamiento, ya que solamente hay un único usuario interactuando con la aplicación al mismo tiempo.

SQLite cuenta con una utilidad por línea de comandos llamada “sqlite” que nos permitirá ejecutar comandos SQL contra una base de datos SQLite en modo consola. Desde aquí podemos crear nuestra base de datos, realizar consultas, insertar datos, etc.

Existen versiones de la API para distintos lenguajes, incluso un proyecto de ODBC para SQLite[URL46].

2.4.3.11 Dalvik

Las aplicaciones para Android se programan en Java, por lo que es necesaria la existencia de una máquina virtual donde se ejecuten las aplicaciones. En el caso de Android, existe una máquina virtual propia llamada Dalvik[URL47].

Dalvik es la máquina virtual que utiliza Android para ejecutar sus aplicaciones, la cual está diseñada y escrita por Dan Bornstein y otros ingenieros de Google. En ella podemos encontrar una gran diferencia con respecto a la máquina virtual Java (JVM), ya que la máquina virtual de Google no está basada en una pila.

Dalvik VM es un intérprete que sólo ejecuta los archivos ejecutables con formato .dex (Dalvik Executable). Este formato está optimizado para el almacenamiento eficiente de la memoria, lo cual se consigue delegando en el kernel la gestión de hilos (multi-threading) tanto de memoria como de procesos.

La herramienta “dx” incluida en el SDK de Android permite transformar las clases compiladas (“.class”) por un compilador de lenguaje Java en formato “.dex”. La Dalvik VM también ha sido optimizada para ejecutar múltiples instancias, ya que cada aplicación se ejecuta en su propia instancia de la máquina virtual.

Principales diferencias con una máquina virtual Java normal

En primer lugar, la máquina virtual de Dalvik toma los archivos generados por las clases Java y los combina en uno o más archivos ejecutables Dalvik (“.dex”), los cuales a su vez son comprimidos en un sólo fichero “.apk” (Android Package) en el dispositivo. De esta forma, reutiliza la información duplicada por múltiples archivos “.class”, reduciendo así la necesidad de espacio (sin comprimir) a la mitad de lo que ocuparía un archivo “.jar”.

Además, la máquina virtual de Dalvik utiliza un tipo diferente de montaje para la generación del código, en el que se utilizan los registros como las unidades primarias de almacenamiento de datos en lugar de la pila. Hay que señalar que el código ejecutable final de Android, como resultado de la máquina virtual de Dalvik, no se basa en el *bytecode* de Java, sino que se basa en los archivos “.dex”. Esto significa que no se puede ejecutar directamente el *bytecode* de Java, sino que



hay que comenzar con los archivos “.class” de Java y luego convertirlos en archivos “.dex”.

Comparativa entre plataformas

A continuación se presenta una tabla comparativa entre las diferentes plataformas evaluadas para la realización de la aplicación que nos ocupa. Recoge diferentes aspectos que puedan ser relevantes o discriminantes a la hora de elegir una plataforma u otra en la realización de la aplicación.

	Windows Phone	iOS	Android
Soporte Flash	No	No	Sí
Soporte HTML5	Sí	Sí	Sí
Almacenamiento externo (tarjeta de memoria)	Sí	No	Sí
Market para aplicaciones	Sí	Sí	Sí
Widgets	Sí	Sí	Sí
Multitáctil	Sí	Sí	Sí
Teclado	Pantalla / Físico	Pantalla / Físico (externo)	Pantalla / Físico
Mapas	Sí	Sí	Sí
Navegación GPS	Sí	Sí	Sí
Integra Base de datos	Sí	Sí	Sí
Hardware Homogéneo	No	Sí	No
Entorno de desarrollo	Windows Phone SDK / Windows	iOS + Cocoa / Mac OS	Android SDK / Windows, Mac OS, Linux
Acceso a internet	Sí	Sí	Sí
Integración de redes sociales	Sí	Sí	Sí

Tabla 15. Comparativa Windows Phone-iOS-Android

Como puede apreciarse en la tabla comparativa, no se encuentra ninguna razón discriminante entre las plataformas que nos indique que no pueda realizarse la aplicación en alguna de ellas.

Por tanto, y sobre todo debido a la naturaleza de la aplicación que pretende fomentar la interacción de usuarios, se propone como escenario ideal la realización de una versión de la aplicación para cada una de las plataformas, aunque en este proyecto solamente se realizará para una única plataforma.

Además, teniendo en cuenta que solamente la plataforma iOS presenta un hardware homogéneo con dispositivos de tamaño y densidad de pantalla conocidos, los tamaños y densidades de pantalla soportados se restringirán para no alargar el desarrollo con elementos que no aportan al presente proyecto, aunque en una aplicación comercial debería tenerse en cuenta.

2.4.3.12 Ventajas y desventajas de solucionar el problema propuesto mediante una aplicación nativa

Ventajas

- Solo usuarios con un terminal que soporte las características de la aplicación pueden acceder a la misma.
- Mayor facilidad para optimizar el consumo de datos mediante sockets.
- Fácil acceso al almacenaje de datos en memoria mediante bases de datos, utilizando el terminal del cliente para almacenar datos relevantes solamente a él, liberando carga en el posible servidor donde almacenamos información de la aplicación.
- Solamente hay que descargar la aplicación una vez y no en cada conexión.
- Al no ser necesario conocer multitud de lenguajes diferentes para el desarrollo (lenguaje pertinente a la plataforma y además XML/JSON para intercambiar datos, etc.) se puede empezar a desarrollar más rápidamente y concentrar esfuerzos en una única tecnología.

Desventajas

- Las actualizaciones han de ser descargadas e instaladas.
- Diferentes usuarios pueden tener diferentes versiones de la aplicación.
- Hay muchos tipos de dispositivos con diferentes capacidades (en menor medida para iOS), por lo que es difícil cubrirlos todos en el desarrollo.

Como se aprecia en esta enumeración de ventajas y desventajas encontradas para una aplicación nativa y en la enumeración de ventajas y desventajas de la solución Web, no hay una solución perfecta, pero para el problema que se pretende solucionar en este proyecto la mejor solución parece ser la de crear una aplicación nativa, ya que puede tratarse de una aplicación con bastante consumo de datos por Internet y que pretende ser utilizada en tabletas. Además, las desventajas halladas no parecen ser un gran impedimento en el desarrollo y aceptación de la aplicación, pues son los mismos problemas a los que se enfrenta cualquier aplicación nativa que hay en el mercado.

Una vez determinado que la opción escogida es la realización de una aplicación nativa, debemos decidir si realizar la aplicación para la plataforma iOS de Apple o para Google Android.

Aunque para iOS solamente existen dos tabletas (iPad de 7,9 y de 9,7 pulgadas) mientras que en Android hay un sin fin de ellas, y a pesar de que los beneficios de comercializar la aplicación pudieran llegar a ser superiores en esta plataforma según se comentó anteriormente en este documento, el desarrollador que debe implementar la solución desconoce el lenguaje y "Framework" de desarrollo para

iOS (Objective-C y Cocoa respectivamente) por lo que habría de realizarse una inversión en tiempo y materiales para su estudio, mientras que para el caso de Java y de Android ya se tienen conocimientos (más de 8 años desarrollando con JAVA SE y una pequeña base previa de desarrollo Android). Además, para cuando sea necesario, se encuentran en la red una gran cantidad de tutoriales gratuitos de desarrollo Android, siendo una fuente de información importante la propia guía de desarrollo que brinda Google (<http://developer.android.com/index.html>), además de existir tutoriales básicos realizados en la propia UC3M, como es el caso del utilizado para la realización de este estudio del estado del arte, perteneciente a la asignatura "Software de comunicaciones" impartida por el departamento de Ingeniería Telemática.

Otros puntos a favor de la elección del sistema Android es su aumento en cuanto a la cuota de mercado y el precio general de sus dispositivos, ya que al tratarse Android de un sistema operativo gratuito, los fabricantes pueden incluirlo libremente en los dispositivos que comercializan, haciendo que exista una gran competencia y que se encuentren tabletas de multitud de precios diferentes, haciéndolas bastante accesibles al consumidor final. Esto es un punto fuerte porque ya se mencionó que la aplicación podría ser utilizada también como complemento al aprendizaje o plataforma para el mismo, por lo que es interesante que los estudiantes puedan acceder a dispositivos con los que ejecutar la aplicación en las aulas o en sus casas.

Por lo tanto, tras el estudio del estado del arte realizado, se decide realizar la aplicación para tabletas que ejecuten el sistema operativo Android de Google, dejando para el futuro su posible adaptación a otras plataformas diferentes.

3. Desarrollo

3.1 Definición del problema

Tal y como se describía al inicio del documento, se pretende desarrollar una aplicación para tabletas que permita reunir diferentes tipos de materiales multimedia, los cuales puedan ser comentados por varios usuarios a la vez que se visualizan.

Tras analizar la situación actual en el "Estado del arte", se ha concluido que a la fecha del inicio del desarrollo de este proyecto, no se han encontrado aplicaciones que cubran esta necesidad y por tanto es interesante proceder a su desarrollo, además, se ha decidido que se va a realizar para dispositivos con sistema operativo Google Android.

Así pues, el objetivo o meta de esta aplicación es que se puedan reunir diferentes usuarios que se comuniquen a través de ella sobre uno o varios materiales multimedia, ya sea un único video alojado en la nube o una colección de varios videos, fotos y/o pases de diapositivas. De este modo, además de modo lúdico, podría utilizarse para su apoyo a conferencias o en el ámbito del e-learning, en MOOCs (Massive Online Open Courses - que en español podría traducirse como "Cursos Masivos de Acceso Online y Gratuito"). Además, la comunicación entre los usuarios se llevará a cabo mediante la utilización de las redes sociales, en este caso, Twitter.

3.2 Metodología a seguir en el desarrollo del proyecto

Para llevar a cabo el desarrollo del proyecto, se va a seguir una metodología de desarrollo ágil, por lo que se obtendrá un *feedback* constante por parte del *cliente* (en este caso, el tutor del proyecto) para definir y evaluar los requisitos del proyecto en cada una de las fases.

El desarrollo se hará en iteraciones (ciclos cortos de desarrollo) y en cada una de ellas hay una serie de pasos que se han de tomar y para el presente proyecto se ha definido el siguiente orden de dichos pasos: Definición de metas generales de la iteración, identificación de requisitos, casos de uso y maquetación inicial de la estética del proyecto o "wireframes".

3.2.1 Metodología ágil en el desarrollo de software

Las metodologías de desarrollo ágiles se basan en el desarrollo iterativo e incremental, donde se hacen pequeñas entregas funcionales periódicamente y el cliente las evalúa proporcionando un *feedback* constante, de manera que los defectos o sugerencias que se incorporan al sistema se van corrigiendo o añadiendo en posteriores iteraciones.

Dichas iteraciones son las unidades en las que se define el proyecto y cada iteración agrupa una o varias funcionalidades del proyecto de manera que el tiempo de desarrollo sea normalmente de entre 1 y 4 semanas[URL48].

Generalmente, cada iteración incluye la planificación, el análisis de requisitos, el diseño, codificación, revisión por parte del cliente y la documentación y al final de la misma, el equipo (normalmente junto al cliente) vuelve a evaluar el estado del proyecto y decide cómo se seguirá avanzando y en qué orden o qué prioridades.

La principal ventaja de las metodologías ágiles del desarrollo de software es que se enfatiza la comunicación directa y periódica tanto entre los integrantes del equipo de desarrollo como con el cliente, pudiendo detectar rápidamente si el proyecto se está adecuando a las necesidades y gustos del cliente y por tanto, permite una importante reducción de costes de desarrollo.

La siguiente imagen, obtenida de Wikipedia[URL49] resume el desarrollo ágil de software.

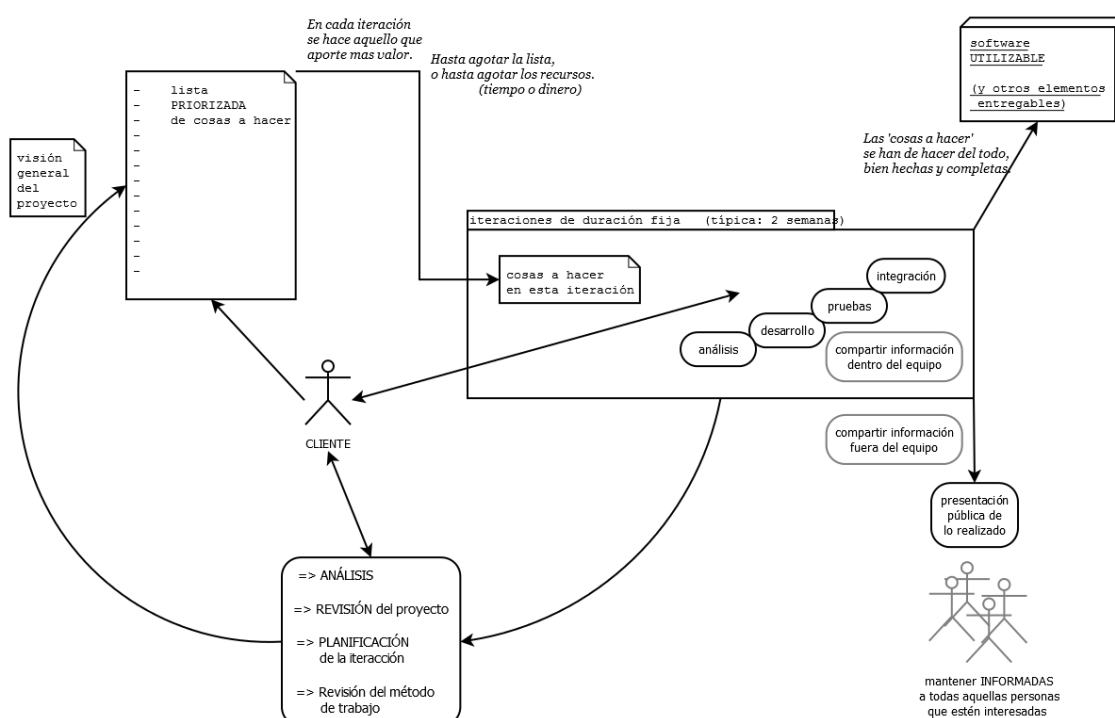


Figura 25. Representación gráfica del desarrollo ágil de software

3.3 Desarrollo del proyecto

A continuación se incluye la documentación generada para las iteraciones que han tenido lugar para el desarrollo del proyecto para las cuales se van a tratar, como mínimo, los siguientes apartados:

3.3.1.1 Metas generales

Recopilará de forma lo más breve posible los objetivos de la iteración.

3.3.1.2 Identificación de requisitos

Requisitos extraídos de las metas generales de la iteración y posibles conversaciones con el cliente del proyecto.

Es importante remarcar que, en siguientes iteraciones, estos requisitos pueden ser refinados o eliminados.

Para la presentación de los mismos, se va a especificar cada requisito utilizando una adaptación simplificada de la plantilla de Volere[URL50] en forma de tabla como la mostrada a continuación, que recoge la información en detalle de los requisitos. Se describe cada uno de los campos que componen esta tabla de plantilla.

R(T)XX			
Título			
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias			
Descripción			
Fuente			

Tabla 16: Plantilla de requisitos del sistema

Donde:

- **R(T)XX:** requisito de tipo (T) número XX. El número de requisito XX ha de ser único y no puede repetirse ningún código de la forma R(T)XX. Si un requisito hubiera de ser eliminado, ese número ya no podría utilizarse.
- **Título:** pequeña descripción general del requisito.
- **Prioridad:** hace referencia al orden temporal en que debe aplicarse el requisito. “Alta” indica lo antes posible y “baja” indica que no es muy urgente.
- **Necesidad:** Importancia o necesidad del requisito en el sistema.
- **Estabilidad:** Indica si el requisito se mantendrá igual a lo largo del proyecto, o si por el contrario podrá ser modificado.

- **Verificable:** Indica si es verificable o no que dicho requisito se ha cumplido
- **Dependencias:** Indica de qué otros requisitos depende o se origina el actual.
- **Descripción:** especificación del requisito mediante texto explicativo.
- **Fuente:** Indica el origen del requisito.

Con el fin de ofrecer una estructura que facilite la comprensión de los requisitos, éstos se van a organizar en varias subcategorías relevantes para este proyecto:

- **Requisitos software funcionales:** especifican qué tiene que hacer el software. Definen el propósito del software. Se les asigna el código “RSF” en las fichas de requisitos.
- **Requisitos software no funcionales:** especifican diferentes factores del software como por ejemplo el cómo ha de ser su interfaz o si ha de cumplir requisitos de seguridad de algún tipo. Vamos a distinguir estos tres tipos en nuestro proyecto:
 - **Requisitos software de comprobación:** especifican las limitaciones que afectan a cómo el software debe verificar los datos de entrada y salida. Se les asigna el código “RSNFC” en las fichas de requisitos.
 - **Requisitos software de restricción:** especifican restricciones que tendrá el sistema. Normalmente son requerimientos del sistema forzados por el cliente o la situación, por ejemplo un tiempo de respuesta máximo, un tamaño mínimo de pantalla soportado o la cantidad máxima de elementos de un tipo a mostrar en una situación. Se les asigna el código “RSNFR” en las fichas de requisitos.
 - **Requisitos software de interfaz de usuario:** especifican requerimientos que ha de tener la interfaz de usuario, ya sean por ejemplo la gama de colores que ha de tener o cómo ha de comportarse una pantalla concreta (es una pantalla de diálogo, se hará con un selector de fecha...), etc. Se les asigna el código “RSNFIdU” en las fichas de requisitos.

3.3.1.3 Casos de uso

Casos de uso de la aplicación a cubrir en la iteración.

3.3.1.4 Wireframes

Wireframes de los casos de uso extraídos para la iteración.

3.3.1.5 Implementación

Comentarios relevantes sobre la implementación de la iteración, por ejemplo, librerías relevantes utilizadas, servicios web consumidos, o decisiones arquitectónicas tomadas.

3.4 Primera iteración

3.4.1 Metas generales

En esta primera iteración se pretende hacer una primera implementación inicial del software, el cual debe comprender una primera versión del total de la funcionalidad requerida por el cliente:

- Creación de nuevos eventos en los que se podrán añadir diferentes materiales multimedia (videos de youtube, presentaciones de Slideshare e imágenes alojadas en Internet)
- Visualización de eventos ya creados, uniéndose mediante un hashtag de evento
- Capacidad de visualizar los materiales a la vez que se pueda seguir la conversación en Twitter en torno al hashtag elegido para el evento

El principal objetivo es que en las iteraciones siguientes solo se incluyan nuevas funcionalidades o mejoras sugeridas por el cliente, siendo la aplicación entregada tras esta iteración una aplicación funcional.

3.4.2 Toma de requisitos

Una vez que nos hemos reunido con el cliente del proyecto, hemos conseguido extraer los requerimientos que el software deberá contemplar. Además de los requisitos de funcionalidad extraídos del cliente, se presentarán requisitos que el equipo de desarrollo cree oportuno tener en cuenta para la implementación, tales como las longitudes mínimas y máximas que los campos de texto tendrán. Por tanto, las fuentes posibles de los requisitos serán: “Cliente” y “Equipo de desarrollo”.

En las páginas siguientes, se detallan los requisitos software del sistema, organizados en las categorías y subcategorías anteriormente descritas.

3.4.3 Identificación de los Requisitos

3.4.3.1 Requisitos de software funcionales

RSF10			
Título	Uso de Twitter en la aplicación		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	La aplicación utilizará la red social Twitter para la comunicación entre sus usuarios.		
Fuente	Cliente		

Tabla 17: Requisito de software funcional 01

RSF20			
Título	Publicar en el Timeline del usuario		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF10		
Descripción	La aplicación deberá ser capaz de publicar en el timeline del usuario en Twitter, utilizando su cuenta.		
Fuente	Cliente		

Tabla 18: Requisito de software funcional 20

RSF30			
Título	Autorización de la aplicación en Twitter		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF20		
Descripción	El usuario no logueado deberá autorizar en su cuenta de Twitter el uso de sus credenciales a través de la aplicación, con permisos para publicar en su nombre.		
Fuente	Cliente		

Tabla 19: Requisito de software funcional 30

RSF40			
Título	Desautorización de la aplicación / Logout		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF30		
Descripción	El usuario logueado deberá ser capaz de eliminar las credenciales que se almacenaron en la aplicación para utilizar su cuenta de Twitter, aunque deberá desvincularla desde la propia web de Twitter.		
Fuente	Cliente		

Tabla 20: Requisito de software funcional 40

RSF50			
Título	Posibles opciones		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	Un usuario logueado podrá crear nuevos eventos o unirse a un evento ya creado.		
Fuente	Cliente		

Tabla 21: Requisito de software funcional 50

RSF60			
Título	Crear nuevos eventos – Datos básicos		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF50		
Descripción	<p>Un usuario logueado podrá crear nuevos eventos. Para ello deberá aportar los siguientes datos:</p> <ul style="list-style-type: none"> • Título del evento • Hashtag del evento (no se podrá utilizar por ningún otro evento en la aplicación) • Fecha de inicio del evento • Fecha de fin del evento (opcional) • Etiquetas para el evento (al menos una) • Descripción del evento • Usuarios a los que avisar de la creación del evento • Elegir si deberá publicarse en Twitter la creación del evento 		
Fuente	Cliente		

Tabla 22: Requisito de software funcional 60

RSF70			
Título	Crear nuevos eventos – Materiales multimedia		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF50		
Descripción	<p>Un usuario logueado, durante el proceso de creación del evento y tras haber introducido los datos básicos de manera correcta, deberá añadir al menos un material multimedia de los siguientes tipos:</p> <ul style="list-style-type: none"> • Video o lista de videos de YouTube • Presentación SlideShare • Imagen <p>Para poder terminar la creación del evento se ha de haber añadido al menos un material de cualquier tipo.</p>		
Fuente	Cliente		

Tabla 23: Requisito de software funcional 70

RSF80			
Título	Crear eventos – Nuevo material multimedia		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF70		
Descripción	Para añadir un material multimedia, éste deberá estar alojado		

	<p>en la nube y el usuario deberá proporcionar los siguientes datos de manera obligatoria:</p> <ul style="list-style-type: none"> • Título del material • URL donde se aloja el material • Descripción del material
Fuente	Cliente

Tabla 24: Requisito de software funcional 80

RSF90	
Título	Unirse a un evento
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF50
Descripción	Un usuario logueado podrá unirse a un evento activo existente en la aplicación.
Fuente	Cliente

Tabla 25: Requisito de software funcional 90

RSF100	
Título	Ver materiales multimedia
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna
Descripción	Un usuario unido a un evento podrá acceder a los materiales multimedia relacionados con el evento.
Fuente	Cliente

Tabla 26: Requisito de software funcional 100

RSF110	
Título	Enviar mensaje al evento
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF10
Descripción	Un usuario unido a un evento podrá enviar un mensaje al evento a través de su cuenta de Twitter. Este mensaje llevará el hashtag definido para el evento durante su creación.
Fuente	Cliente

Tabla 27: Requisito de software funcional 110

RSF120			
Título	Visualizar mensajes relacionados al evento		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF10		
Descripción	Un usuario unido a un evento podrá ver mensajes que se publiquen en la red social Twitter con el hashtag elegido para el evento, independientemente de la aplicación utilizada para su publicación y pudiendo darse el caso de ver mensajes etiquetados con dicho hashtag y que no estén directamente relacionados con el evento.		
Fuente	Cliente		

Tabla 28: Requisito de software funcional 120

RSF130			
Título	Visualizar mensajes y multimedia al mismo tiempo		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF100, RSF120		
Descripción	Un usuario unido a un evento podrá ver mensajes del evento al mismo tiempo que ve uno de los materiales multimedia.		
Fuente	Cliente		

Tabla 29: Requisito de software funcional 130

3.4.3.2 Requisitos de software no funcionales

Requisitos de comprobación

RSNFC10			
Título	Longitud mínima del título de un evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60		
Descripción	Un título de evento tendrá un mínimo de 20 caracteres.		
Fuente	Equipo de desarrollo.		

Tabla 30: Requisito de Software no funcional de comprobación 10

RSNFC20			
Título	Longitud máxima del título de un evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60		
Descripción	Un título de evento tendrá un máximo de 250 caracteres.		
Fuente	Equipo de desarrollo.		

Tabla 31: Requisito de Software no funcional de comprobación 20

RSNFC30			
Título	Longitud máxima del hashtag de un evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60		
Descripción	La longitud máxima del hashtag de un evento será de 25 caracteres incluyendo el símbolo #.		
Fuente	Equipo de desarrollo		

Tabla 32: Requisito de Software no funcional de comprobación 30

RSNFC40			
Título	Longitud mínima del hashtag de un evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60		
Descripción	La longitud mínima del hashtag de un evento será de 2 caracteres incluido el símbolo "#", por lo tanto, deberá introducirse al menos un carácter por parte del usuario.		
Fuente	Cliente		

Tabla 33: Requisito de Software no funcional de comprobación 40

RSNFC50	
Título	Fecha mínima para finalización de evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	La fecha de fin de un evento ha de ser igual o posterior a la de inicio del evento.
Fuente	Cliente

Tabla 34: Requisito de Software no funcional de comprobación 50

RSNFC60	
Título	Número mínimo de etiquetas de un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	Un evento ha de tener al menos una etiqueta.
Fuente	Cliente

Tabla 35: Requisito de Software no funcional de comprobación 60

RSNFC70	
Título	Longitud mínima del campo etiquetas
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	El campo etiquetas ha de tener al menos 1 caracter.
Fuente	Cliente

Tabla 36: Requisito de Software no funcional de comprobación 70

RSNFC80	
Título	Longitud máxima del campo etiquetas
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	La longitud máxima del campo etiquetas ha de ser de 250 caracteres incluyendo espacios y comas.
Fuente	Equipo de desarrollo.

Tabla 37: Requisito de Software no funcional de comprobación 80

RSNFC90	
Título	Opcionalidad del campo “notificar a”
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	Este campo es opcional, pudiendo estar vacío.
Fuente	Cliente

Tabla 38: Requisito de Software no funcional de comprobación 90

RSNFC100	
Título	Formato de usuarios a notificar
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSNFC90
Descripción	Los usuarios introducidos en el campo “notificar a” en la creación del evento estarán separados por comas, su primer carácter será la arroba “@” y no contendrán espacios interiores (por ejemplo, no es posible un usuario como: “user” o como “@un usuario” pero sí “@unusuario”). La longitud mínima de caracteres de cada usuario será de 2 caracteres y el máximo de 15.
Fuente	Cliente, http://support.twitter.com/entries/14609

Tabla 39: Requisito de Software no funcional de comprobación 100

RSNFC110	
Título	Longitud mínima de la descripción de un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	La descripción de un evento tendrá al menos 20 caracteres.
Fuente	Equipo de desarrollo.

Tabla 40: Requisito de Software no funcional de comprobación 110

RSNFC120	
Título	Longitud máxima de la descripción de un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF60
Descripción	La longitud máxima de la descripción de un evento es de 65000 caracteres.
Fuente	Equipo de desarrollo.

Tabla 41: Requisito de Software no funcional de comprobación 120

RSNFC130	
Título	Número máximo de videos de YouTube en un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF70
Descripción	El número máximo de videos de YouTube a añadir en un evento será de 5.
Fuente	Cliente

Tabla 42: Requisito de Software no funcional de comprobación 130

RSNFC140	
Título	Número máximo de presentaciones de Slideshare en un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF70
Descripción	El número máximo de presentaciones de Slideshare a añadir en un evento será de 5.
Fuente	Cliente

Tabla 43: Requisito de Software no funcional de comprobación 140

RSNFC150	
Título	Número máximo de imágenes en un evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF70
Descripción	El número máximo de imágenes en un evento será de 5.
Fuente	Cliente

Tabla 44: Requisito de Software no funcional de comprobación 150

RSNFC160	
Título	Longitud máxima del título de un material multimedia
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF80
Descripción	El título de un material multimedia tendrá un máximo de 250 caracteres.
Fuente	Equipo de desarrollo.

Tabla 45: Requisito de Software no funcional de comprobación 160

RSNFC170	
Título	Longitud mínima del título de un material multimedia
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF80
Descripción	El título de un material multimedia tendrá un mínimo de 20 caracteres.
Fuente	Equipo de desarrollo.

Tabla 46: Requisito de Software no funcional de comprobación 170

RSNFC180	
Título	Longitud máxima de la descripción de un material
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF80
Descripción	La descripción de un material multimedia tendrá un máximo de 500 caracteres.
Fuente	Equipo de desarrollo.

Tabla 47: Requisito de Software no funcional de comprobación 180

RSNFC190	
Título	Longitud mínima de la descripción de un material
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF80
Descripción	La descripción de un material multimedia tendrá un mínimo de 20 caracteres.
Fuente	Equipo de desarrollo.

Tabla 48: Requisito de Software no funcional de comprobación 190

RSNFC200	
Título	Longitud máxima de los mensajes emitidos durante un evento
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF110
Descripción	La longitud máxima de los mensajes a enviar en un evento será de 140 caracteres menos los que ocupe el hashtag elegido para el evento incluyendo el símbolo "#".
Fuente	Twitter, Cliente

Tabla 49: Requisito de Software no funcional de comprobación 200

Requisitos de restricción

RSNFR10			
Título	Sistema operativo soportado		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	La aplicación será desarrollada para funcionar en dispositivos con sistema operativo Android 4.0 en adelante.		
Fuente	Cliente		

Tabla 50: Requisito de Software no funcional de restricción 10

RSNFR20			
Título	Aplicación Horizontal		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	La aplicación tendrá que ser desarrollada para poder utilizarse con el dispositivo en posición horizontal, es decir, el ancho de la pantalla será mayor que su altura.		
Fuente	Cliente		

Tabla 51: Requisito de Software no funcional de restricción 20

RSNFR30			
Título	Tamaño de los dispositivos soportados		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	El tamaño mínimo de los dispositivos soportados será de 7 pulgadas de diagonal de pantalla y el máximo, de 10.1 pulgadas de diagonal de pantalla.		
Fuente	Cliente		

Tabla 52: Requisito de Software no funcional de restricción 30

RSNFR40			
Título	Densidad de pantalla soportada		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	La aplicación soportará dispositivos con una densidad de pantalla media, es decir, de 150 a 200 dpi.		
Fuente	Cliente		

Tabla 53: Requisito de Software no funcional de restricción 40

RSNFR50			
Título	Frecuencia de refresco de mensajes de un evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	Los tuits a mostrar durante un evento se actualizarán cada 10 segundos		
Fuente	Equipo de desarrollo.		

Tabla 54: Requisito de Software no funcional de restricción 50

Requisitos de interfaz de usuario

RSNFIdU10			
Título	Visibilidad simultánea de opciones en evento		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF100, RSF120, RSF130		
Descripción	Se han de mostrar en una misma pantalla los materiales disponibles en el evento, el material visionado y el chat.		
Fuente	Cliente		

Tabla 55: Requisito de Software no funcional de interfaz de usuario 10

RSNFIdU20			
Título	Menú deslizante lateral		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	El menú que contiene los materiales disponibles para un evento estará en un menú deslizante situado a la izquierda de la pantalla y podrá mostrarse y ocultarse.		
Fuente	Cliente		

Tabla 56: Requisito de Software no funcional de interfaz de usuario 20

RSNFIdU30			
Título	Chat en la parte derecha de la pantalla		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	El chat del evento mostrará sus mensajes en la parte derecha de la pantalla.		
Fuente	Cliente		

Tabla 57: Requisito de Software no funcional de interfaz de usuario 30

RSNFIdU40			
Título	Posición de material multimedia		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	El material quedará situado a la izquierda de la aplicación, ocupando todo el ancho excepto el espacio necesario para el chat. Cuando el menú lateral deslizante esté activo, el material multimedia activo y el chat quedarán en segundo plano.		
Fuente	Equipo de desarrollo.		

Tabla 58: Requisito de Software no funcional de interfaz de usuario 40

RSNFIdU50			
Título	Aviso de errores en campos de texto		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	Cuando haya un error en un campo de texto, se mostrará el error con un mensaje junto al campo con dicho error.		
Fuente	Cliente		

Tabla 59: Requisito de Software no funcional de interfaz de usuario 50

RSNFIdU60			
Título	Wizard de creación de eventos		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna		
Descripción	Para la creación de eventos, se solicitarán los datos de forma gradual en un “Wizard” que permita la navegabilidad hacia delante y hacia detrás entre las diferentes pantallas de creación.		
Fuente	Cliente		

Tabla 60: Requisito de Software no funcional de interfaz de usuario 60

3.4.4 Casos de uso

Para poder diagramar los casos de uso de una manera sencilla e inteligible se han decidido crear cuatro tipos de actores en el sistema:

Usuario no logueado: Representa un usuario que no ha autorizado con éxito el uso de la aplicación en Twitter a través del proceso *oAuth* o bien a uno que lo hizo pero que ha optado por salir la aplicación, borrando sus credenciales de la misma.

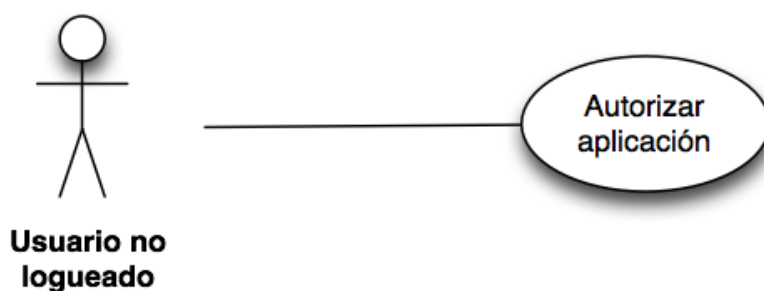


Figura 26. Caso de uso para usuario no logueado

Usuario logueado: Representa a un usuario que ha autorizado con éxito el uso de la aplicación en su cuenta de Twitter mediante el proceso de autenticación *oAuth*. Se van a distinguir dos subtipos de usuario logueado: "Usuario creando evento" y "usuario en evento"

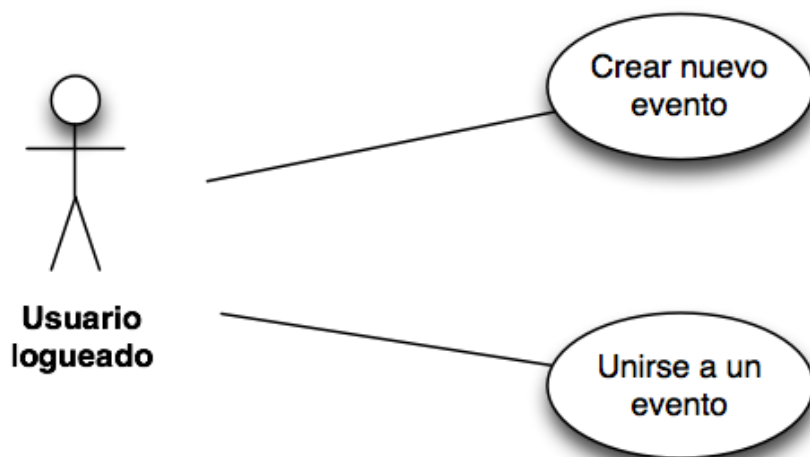


Figura 27. Caso de uso para usuario logueado

Usuario creando un evento: Representa a un usuario logueado que está creando un nuevo evento.

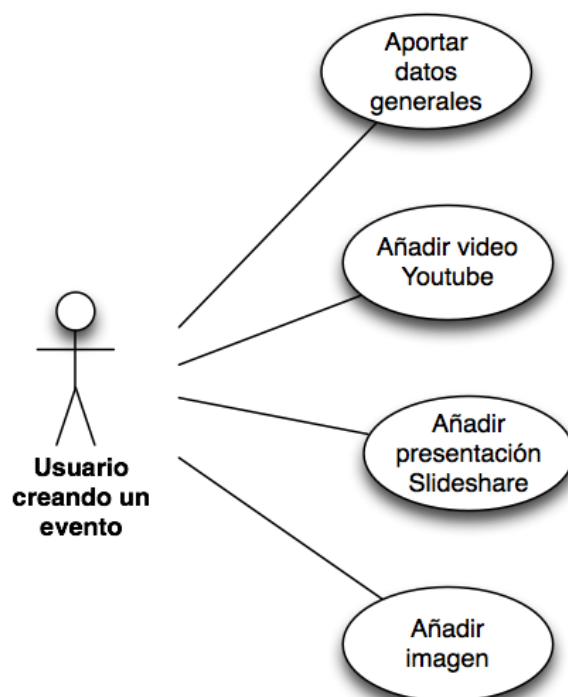


Figura 28. Caso de uso para usuario asistiendo a evento

Usuario asistiendo a un evento: Representa a un usuario logueado que está siguiendo un evento ya existente, es decir, se encuentra en la pantalla de evento.

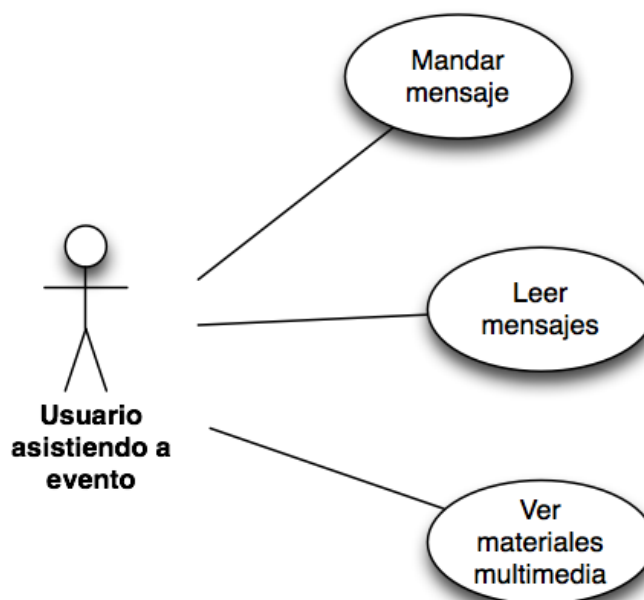


Figura 29. Caso de uso para usuario creando un evento

3.4.5 Wireframes

Una vez definidos tanto los requisitos del sistema como sus casos de uso, estamos en condiciones de hacer una maqueta preliminar de la aplicación. Se ha realizado una primera maqueta a mano presentada al cliente (en el “Anexo A” Se pueden observar los *wireframes* presentados al cliente como primer acercamiento, sobre los cuales también se hizo un recorte de la funcionalidad exigida) y una vez estudiados dichos bocetos por el cliente y en una reunión posterior, se han determinado una serie de requisitos y se tiene una primera versión inicial sobre la que empezar a trabajar.

La apariencia aproximada acordada con el cliente será como se refleja en los bocetos que pueden verse a continuación y que se ha separado en dos bloques principales, creación de nuevos eventos y visualización de un evento.

3.4.5.1 Wireframes para la creación de nuevos eventos

Pantalla inicial de la aplicación

Se trata de la pantalla que aparece cuando iniciamos la aplicación, la cual no tiene permisos para interactuar con una cuenta de Twitter.



Figura 30. Boceto de imagen inicial de la aplicación para autorizar el uso de la cuenta de Twitter del usuario

Pantalla de opciones

Pantalla principal de la aplicación que aparece cuando ésta ya tiene permiso para utilizar una cuenta de Twitter. Desde aquí se puede elegir lo que se va a hacer: Unirse a un evento o crear uno nuevo.

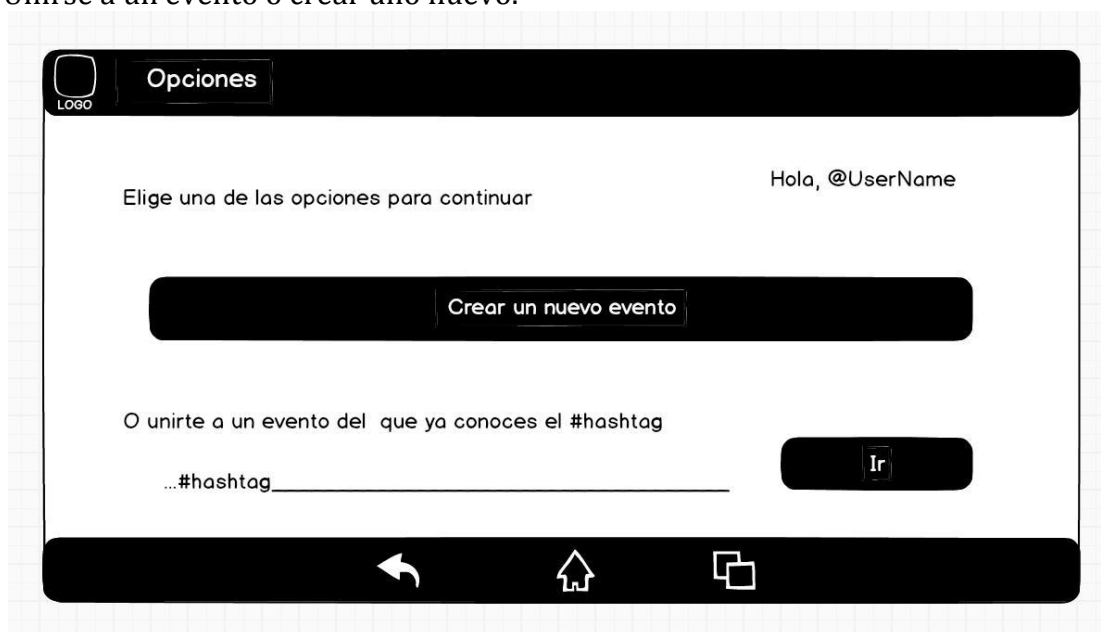


Figura 31. Boceto de la pantalla de opciones de la aplicación

Pantallas para la creación de un evento

La creación de eventos se ha de separar en tres pantallas diferentes para evitar pedir una gran cantidad de información al usuario de una sola vez. Estas pantallas o pasos han de ser navegables entre sí, es decir, estando en el paso 2 se deberá poder volver al paso 1 o ir al paso 3 si los datos introducidos son correctos. Los bocetos resultantes para estas pantallas pueden verse a continuación:

Paso 1: En este primer paso de la creación requerirán los datos más esenciales del evento.



Crear Evento Paso 1

Nombre del Evento: _____

Hashtag para el Evento: _____

Fecha de inicio: _____

Fecha de fin: _____

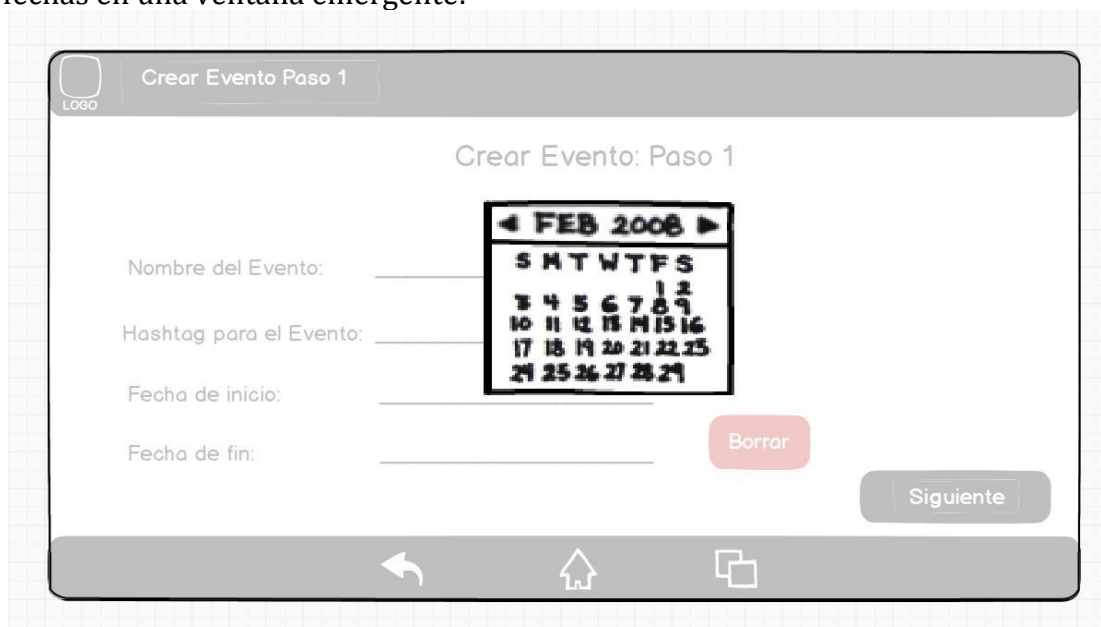
Borrar

Siguiente

Figura 32. Boceto del primer paso en la creación de eventos

El nombre del evento, el hashtag y la fecha de inicio serán obligatorios. El botón “Borrar” anulará la fecha de fin introducida en caso de haberse introducido, ya que esta es opcional. Al pulsar el botón “Siguiente” se validará que los campos tengan un formato correcto y que el hashtag introducido aún no se utiliza en otro evento de la aplicación.

Para introducir las fechas se utilizará algún tipo de calendario o selector de fechas en una ventana emergente:



Crear Evento Paso 1

Nombre del Evento: _____

Hashtag para el Evento: _____

Fecha de inicio: _____

Fecha de fin: _____

Borrar

Siguiente

Calendar overlay: FEB 2008

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	

Figura 33. Boceto del proceso de selección de fechas durante la creación de un evento

Mientras se validan los datos, se informará al usuario de ello y en caso de error, se informará también.



Figura 34. Boceto de feedback al usuario mientras se validan los datos

Cuando exista un error en un campo, se informará de ello y se ayudará a identificar el campo donde se dio el error y la causa y cuando el hashtag esté ocupado, se informará al usuario con una ventana emergente:

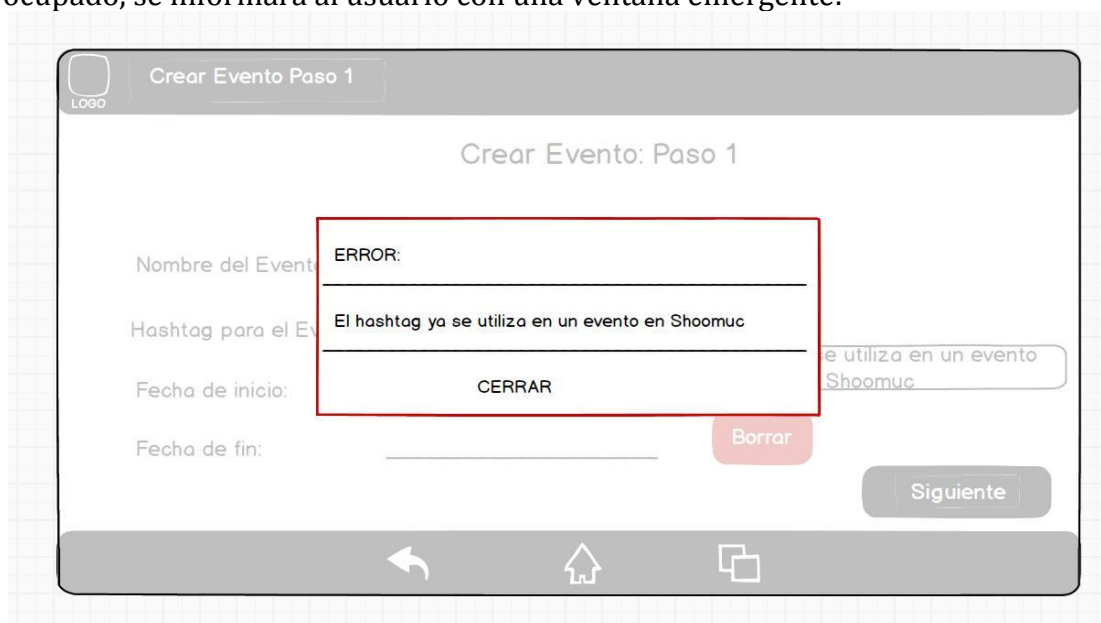
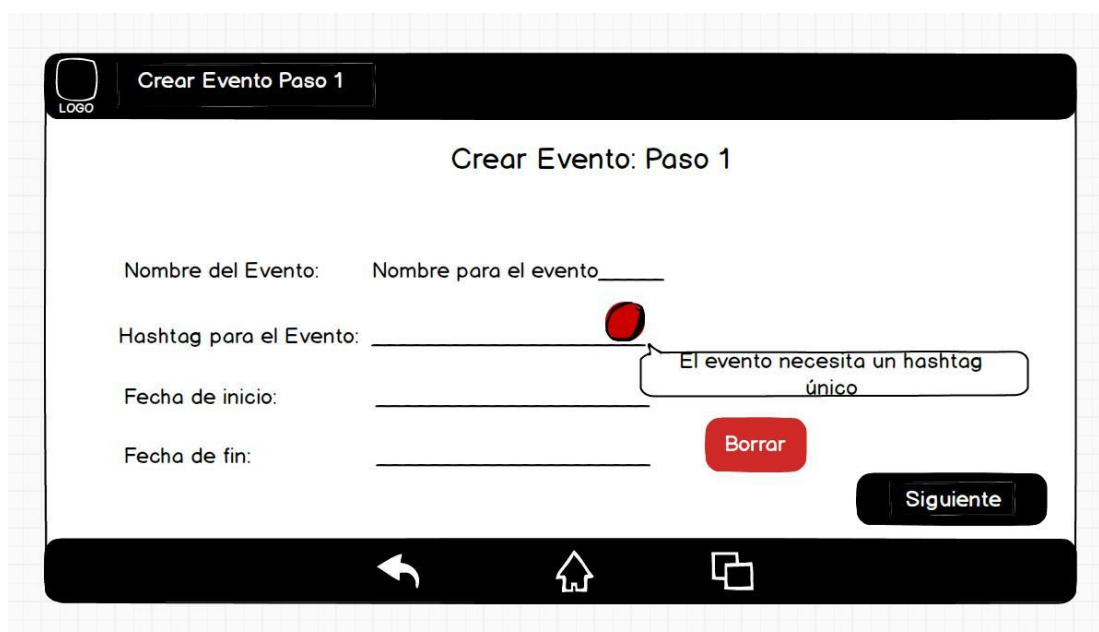


Figura 35. Boceto de error cuando el hashtag ya ha sido usado en la aplicación



Crear Evento Paso 1

Nombre del Evento: Nombre para el evento _____

Hashtag para el Evento: ● El evento necesita un hashtag único

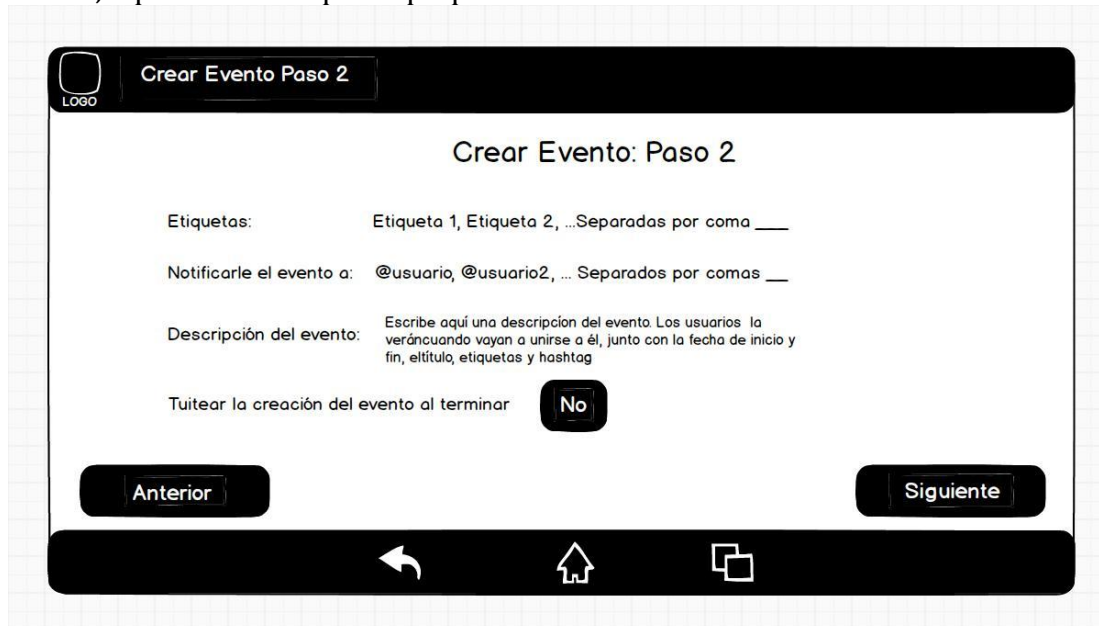
Fecha de inicio:

Fecha de fin:

Borrar Siguiente

Figura 36. Boceto de error porque el hashtag no se ha introducido

Paso 2: Se requerirá al usuario que introduzca información complementaria del evento, aquellos datos que impliquen describir el evento o darlo a conocer.



Crear Evento Paso 2

Etiquetas: Etiqueta 1, Etiqueta 2, ...Separadas por coma ____

Notificarle el evento a: @usuario, @usuario2, ... Separados por comas ____

Descripción del evento: Escribe aquí una descripción del evento. Los usuarios la verán cuando vayan a unirse a él, junto con la fecha de inicio y fin, el título, etiquetas y hashtag

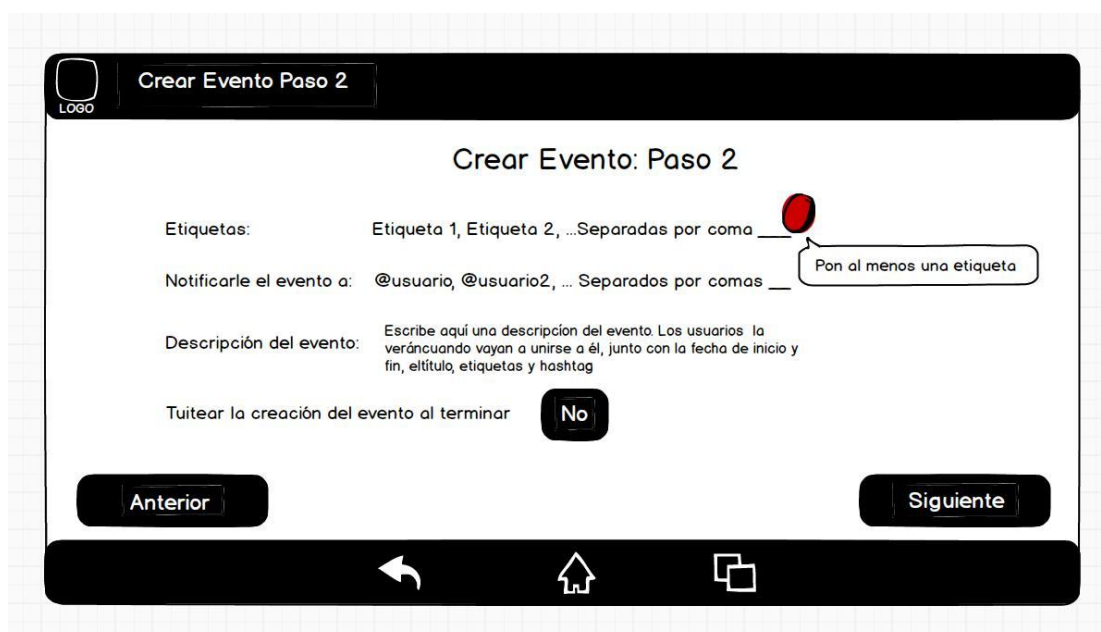
Tuitear la creación del evento al terminar No

Anterior Siguiente

Figura 37. Boceto del paso 2 de la creación de eventos

El usuario deberá introducir al menos una etiqueta y la descripción del evento. Los usuarios a notificar son opcionales.

Los errores se informarán con los mismos criterios que en el paso anterior:



Logo

Crear Evento Paso 2

Crear Evento: Paso 2

Etiquetas: Etiqueta 1, Etiqueta 2, ...Separadas por coma

Notificarle el evento a: @usuario, @usuario2, ... Separados por comas

Descripción del evento: Escribe aquí una descripción del evento. Los usuarios la verán cuando vayan a unirse a él, junto con la fecha de inicio y fin, el título, etiquetas y hashtag

Tuitear la creación del evento al terminar

Navigation icons: back, home, tabs

Error message: Pon al menos una etiqueta

Figura 38. Boceto de información de error al usuario porque no ha etiquetado el evento



Logo

Crear Evento Paso 2

Crear Evento: Paso 2

Etiquetas: Etiqueta 1, Etiqueta 2, ...Separadas por coma

Notificarle el evento a: @usuario, sinArroba, ... Separados por comas

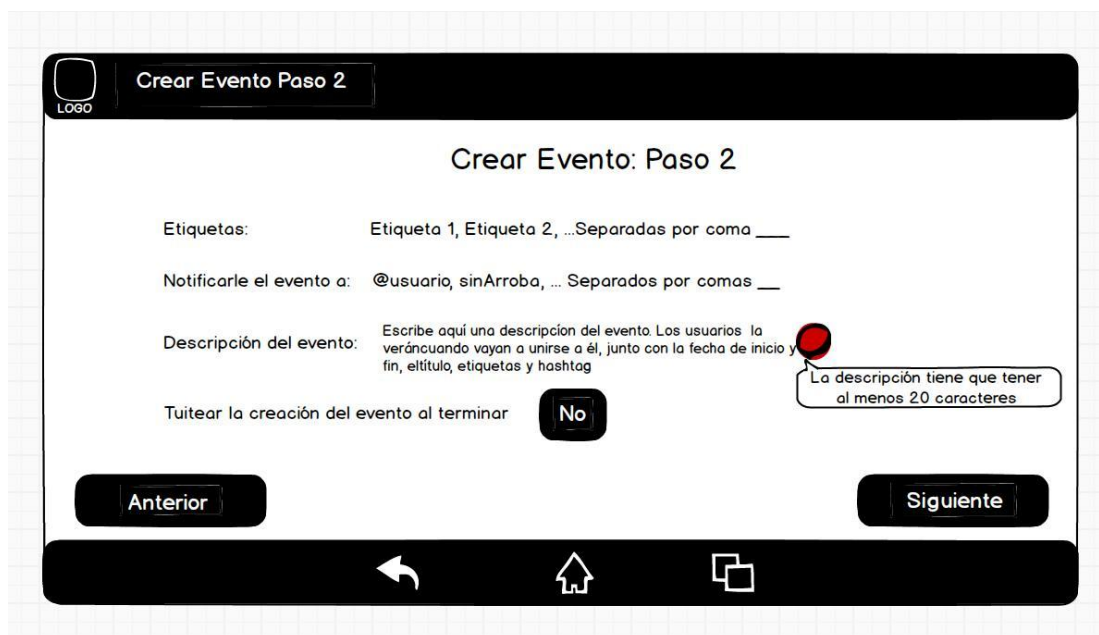
Descripción del evento: Escribe aquí una descripción del evento. Los usuarios la verán cuando vayan a unirse a él, junto con la fecha de inicio y fin, el título, etiquetas y hashtag

Tuitear la creación del evento al terminar

Navigation icons: back, home, tabs

Error message: Hay un error en los usuarios a notificar: sinArroba

Figura 39. Boceto de información de error al usuario porque no ha facilitado los usuarios a notificar con un formato correcto



Crear Evento Paso 2

Etiquetas: Etiqueta 1, Etiqueta 2, ...Separadas por coma ____

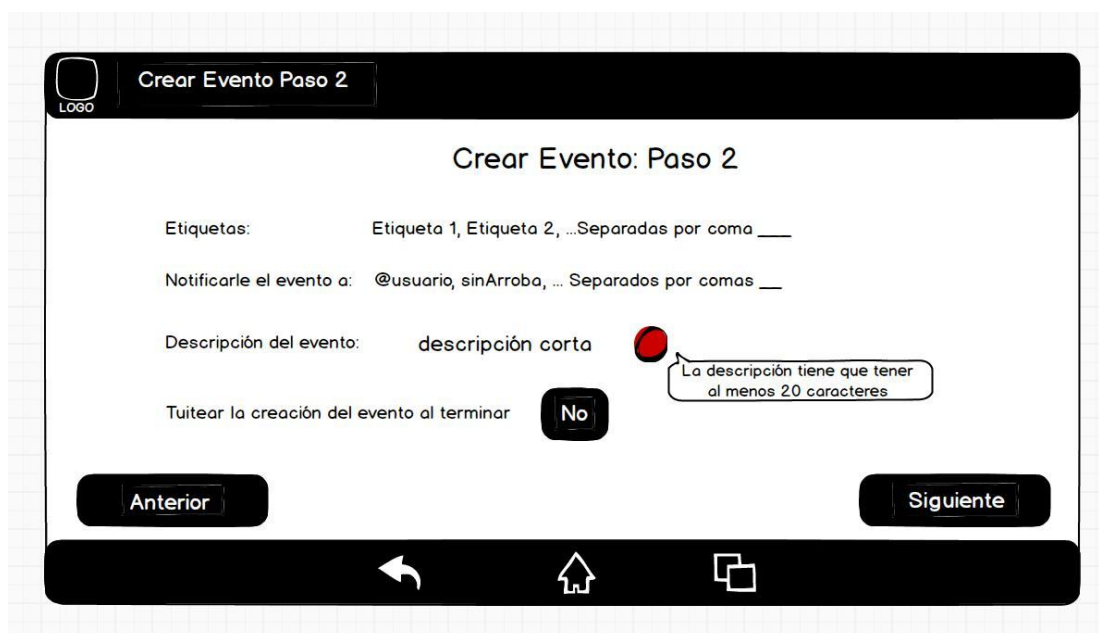
Notificarle el evento a: @usuario, sinArroba, ... Separados por comas ____

Descripción del evento: Escribe aquí una descripción del evento. Los usuarios la verán cuando vayan a unirse a él, junto con la fecha de inicio y fin, el título, etiquetas y hashtag

Tuitear la creación del evento al terminar

La descripción tiene que tener al menos 20 caracteres

Figura 40. Boceto de notificación de error cuando el usuario no pone una descripción de evento



Crear Evento Paso 2

Etiquetas: Etiqueta 1, Etiqueta 2, ...Separadas por coma ____

Notificarle el evento a: @usuario, sinArroba, ... Separados por comas ____

Descripción del evento: descripción corta

Tuitear la creación del evento al terminar

La descripción tiene que tener al menos 20 caracteres

Figura 41. Boceto de notificación de error cuando el usuario pone una descripción de evento demasiado corta

Paso 3: Se solicitará al usuario que introduzca los materiales multimedia soportados por la aplicación, con un máximo de 5 materiales de cada tipo.

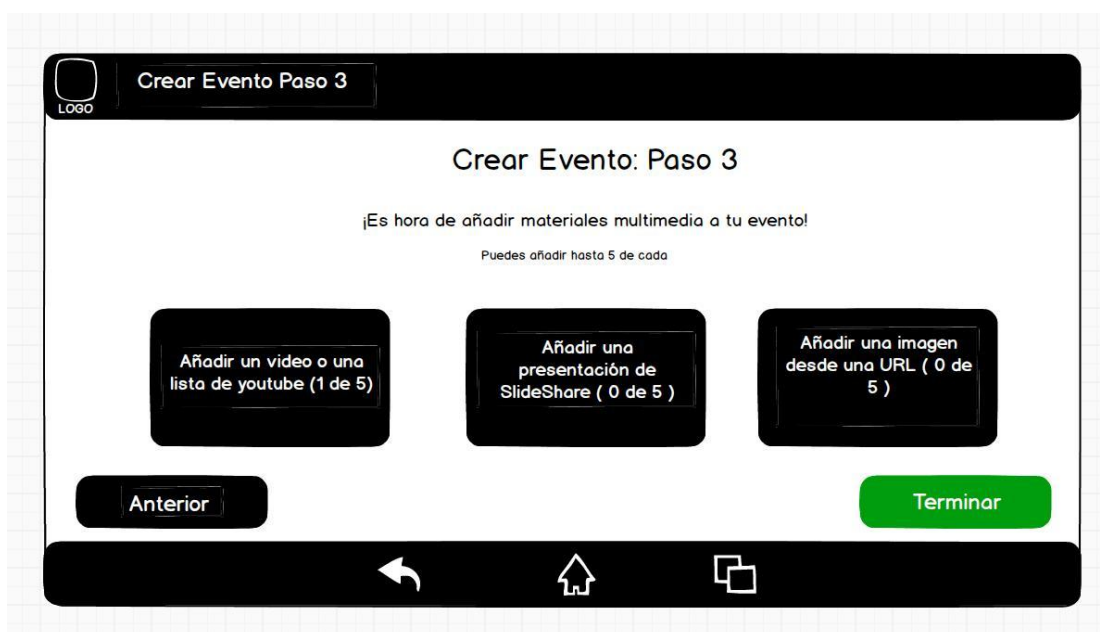


Figura 42. Boceto del Paso 3 de creación de evento

Para terminar la creación del evento se deberá introducir al menos un material del tipo que sea.

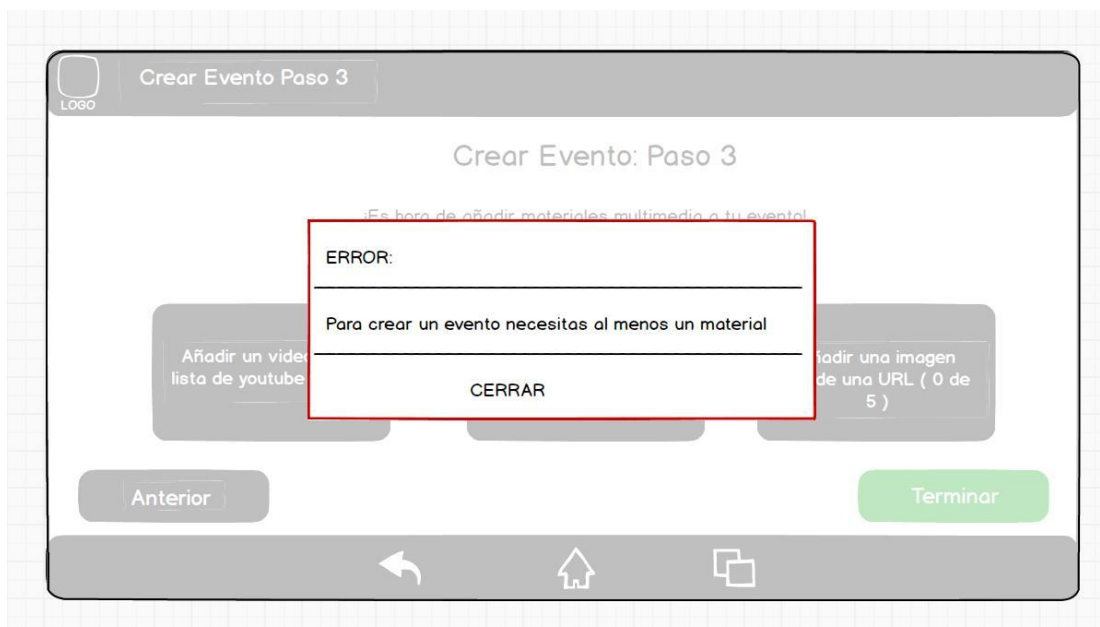


Figura 43. Boceto de notificación de error porque no se han introducido materiales

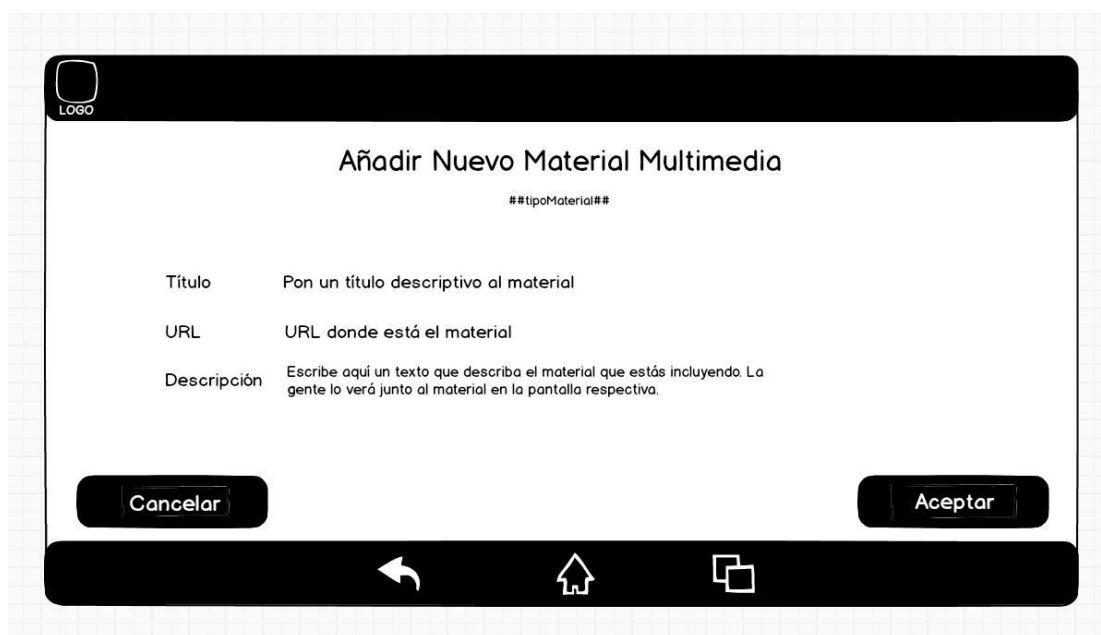


Figura 44. Boceto de pantalla de adición de nuevo material

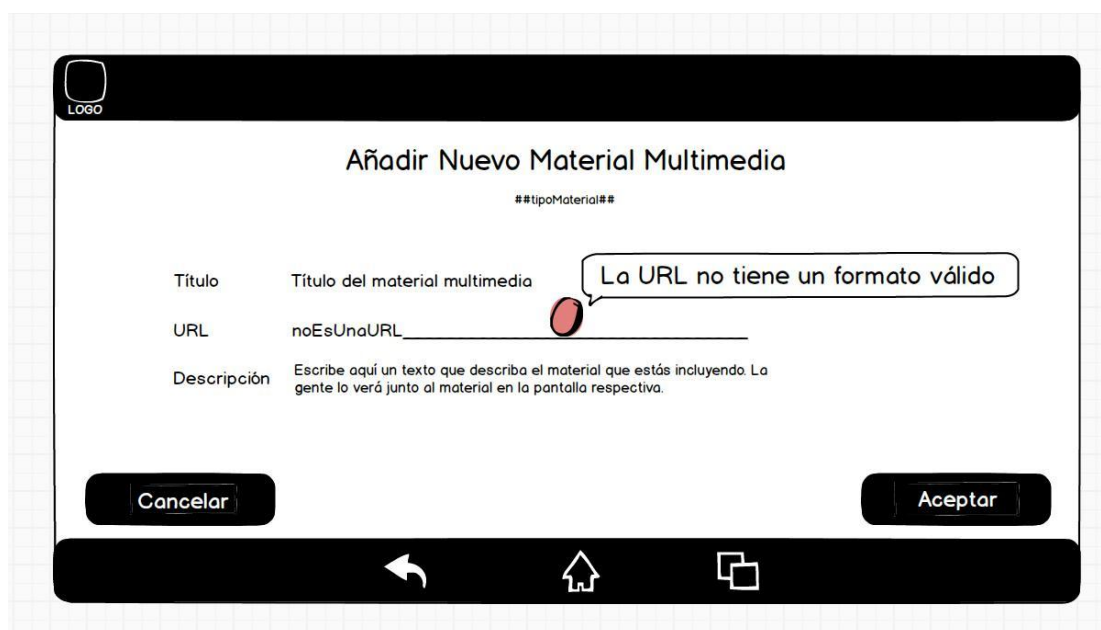


Figura 45. Boceto de error en el formato de la URL



Figura 46. Boceto de notificación de evento creado con éxito

3.4.5.2 Wireframes para la visualización de un evento

La visualización de un evento solamente constará de una pantalla, la cual poseerá un menú lateral deslizante para poder cambiar entre los diferentes materiales.

En la figura 47 se puede ver cómo habrá de ser la pantalla de visualización del evento:

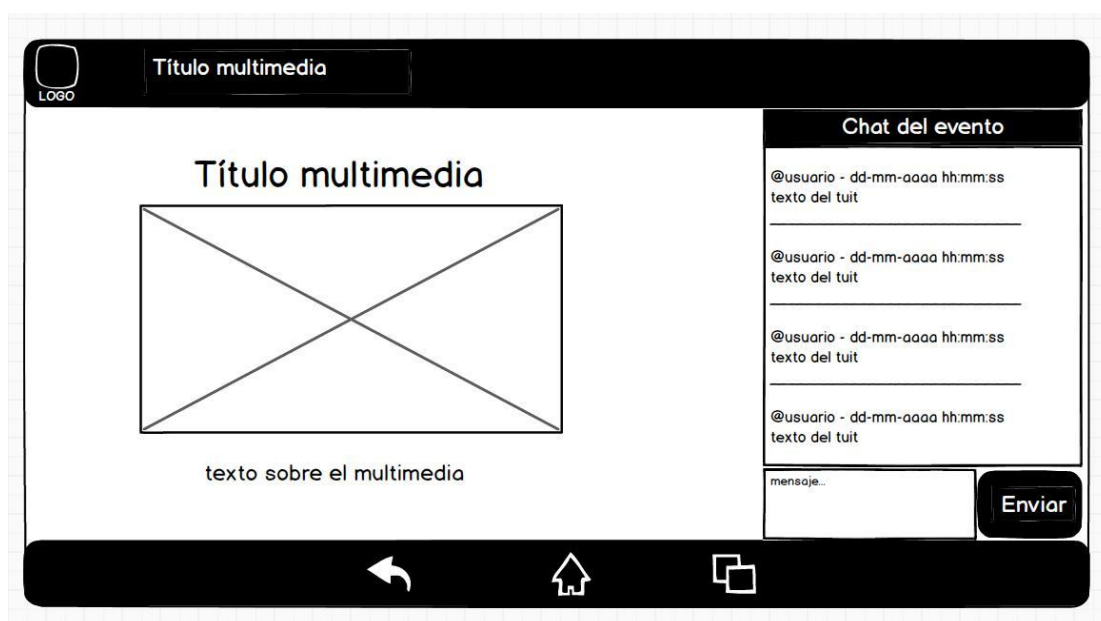


Figura 47. Boceto de visualización de evento

Y en la siguiente figura, la figura 48 se puede ver la apariencia aproximada que tendrá la pantalla de visualización de evento cuando se despliegue el menú lateral, donde se podrá elegir el material a cargar.

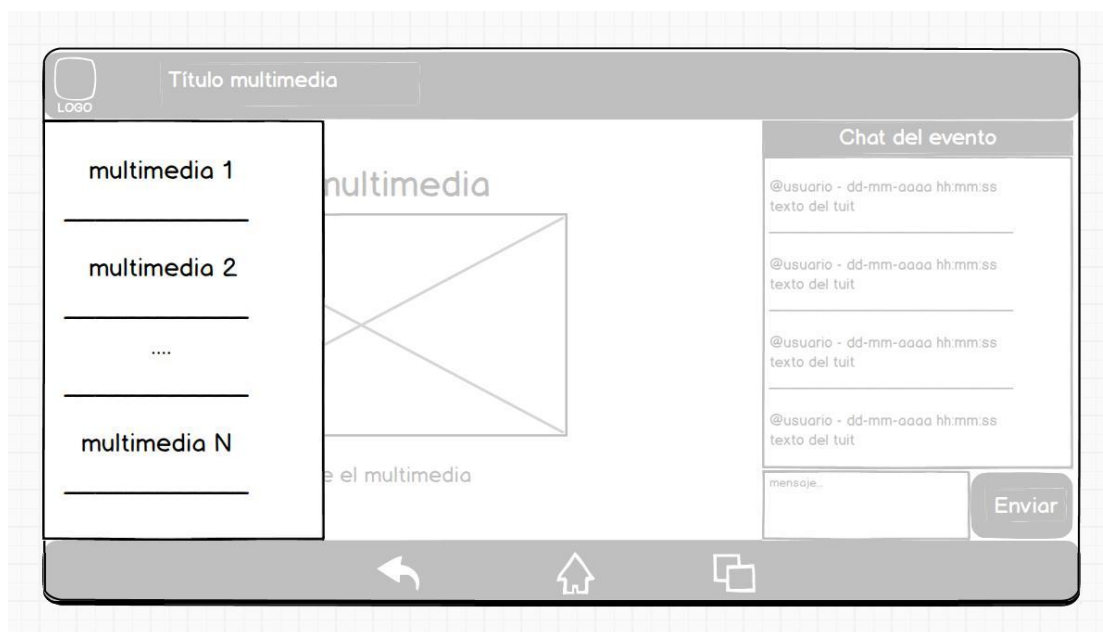


Figura 48. Boceto de visualización de evento con menú lateral desplegado

3.4.6 Implementación

Al tratarse de una iteración inicial del proceso de desarrollo, es conveniente plasmar una serie de decisiones de implementación tomadas y que son relevantes de cara al resto del desarrollo y mantenimiento del software. Estos puntos a tratar en esta iteración son los siguientes:

- Arquitectura elegida
- Librerías relevantes elegidas: Twitter4j
- Sistema de autenticación en Twitter (oAuth)
- Datos devueltos por Twitter en las peticiones de búsqueda. Campos relevantes.
- Obtención de código para el empotrado en la aplicación de los videos de youtube y presentaciones de Slideshare a partir de la URL proporcionada.
- API creada para obtener información de los eventos y almacenarlos.
- Decisiones de diseño impuestas por terceros

3.4.6.1 Arquitectura elegida

Al tratarse de un requisito esencial el que los materiales multimedia utilizados en el evento sean alojados por terceros y el hecho de que múltiples personas a la vez puedan acceder a los mismos contenidos de manera simultánea, se ha elegido la arquitectura “**Cliente/Servidor**”, donde **la aplicación será el cliente** que consumirá datos devueltos por diferentes servicios web, ya sean de creación propia o de terceros, como Twitter o el servicio sobre formato oEmbed de Youtube o Slideshare. Esto es importante, ya que implica que la aplicación servirá para convertir la tableta en un terminal de consulta y que no almacenará información relativa a los eventos en el mismo, accediendo cada vez al material publicado actualmente.

Habrà además un servidor encargado de alojar la base de datos para la aplicación así como de exponer una API para interactuar con la base de datos, como se explicará en otros apartados

3.4.6.2 Librerías utilizadas

Twitter4j (<http://twitter4j.org/en/index.html>) es una herramienta Java implementada en 2007 por Yusuke Yamamoto que permite facilitar la integración con Twitter, permitiendo incluir funcionalidades asociadas a esta red social en nuestra aplicación de una manera más sencilla[URL51].

Esta herramienta permite interactuar con Twitter para gestionar tuits, usuarios, listas, mensajes directos, relaciones, favoritos, subscripciones, bloqueos, realizar búsquedas, etc.

Las características principales de Twitter4j son:

- Compatibilidad con Java 5 o superior.
- Compatibilidad para plataforma Android, por estar implementado en Java.
- No existen dependencias con librerías externas.
- Soporte del estándar OAuth para la autenticación.

Para la realización de cualquier operación, previamente es preciso validarse a través de las clases de autenticación y credenciales. Una vez conseguida la autenticación, es posible realizar las operaciones que el usuario validado tenga permitido llevar a cabo.

A la fecha de este escrito, la versión actual publicada y que se utilizará es Twitter4j v3.0.3 que se publica bajo licencia Apache 2.0[URL52].

3.4.6.3 Sistema de autenticación en Twitter (OAuth)

Para poder interactuar con la aplicación Twitter desde aplicaciones externas como la que se va a desarrollar, es necesario que el usuario autorice la aplicación, lo que se hará mediante el protocolo OAuth [URL53].

OAuth (Open Authorization) es un protocolo abierto y originado, precisamente, por los colaboradores de Twitter Blaine Cook y Chris Messina, el cual permite tener una autorización que asegure que quien accede a un cierto servicio es quien dice ser, aunque lo haga desde clientes externos a quienes proveen dicho servicio.

OAuth puede entenderse desde el punto de vista del consumidor y del punto de vista del proveedor del servicio. Para el consumidor, OAuth es un método de interactuar con datos protegidos y publicarlos. Para el proveedor del servicio, OAuth proporciona a los usuarios un acceso a sus datos al mismo tiempo que protege las credenciales de su cuenta. En otras palabras, OAuth permite a un usuario del sitio/aplicación “A” compartir su información alojada en dicho sitio/aplicación (proveedor de servicio) con el sitio/aplicación B (llamado consumidor) sin la necesidad compartir toda su identidad.

El procedimiento a seguir por nuestra aplicación será el siguiente:

1. Se registrará la aplicación (una única vez y de forma manual) en la sección para desarrolladores de Twitter, introduciendo un nombre, descripción, permisos requeridos y una URL a la que Twitter deberá redirigirse proveyendo la respuesta del usuario denominada *callback*, lo cual nos generará unas credenciales de aplicación conocidas como “*Consumer key*” y “*Consumer Secret*”. Estas credenciales se utilizarán para firmar las peticiones hechas desde nuestra aplicación, de manera que Twitter pueda verificar el origen.

2. El usuario deberá conceder autorización a nuestra aplicación en su cuenta de Twitter, para ello se le redirigirá a la URL de autenticación OAuth de Twitter junto con la firma de nuestra aplicación para que Twitter pueda identificar la aplicación a la que el usuario deberá otorgar acceso.
3. El usuario deberá introducir su nombre de usuario y su contraseña de Twitter en la web de Twitter a la que se le redirigió, conceder acceso a su cuenta a dicha aplicación con los permisos que esta requiere y que le son debidamente informados.
4. Si estos datos son válidos, Twitter redirigirá al usuario de vuelta a la URL de *callback* proporcionada al registrar la aplicación en Twitter, adjuntando en su respuesta los valores de las credenciales ("*Consumer key*" y "*Consumer secret*") del usuario. Estas credenciales serán utilizadas como autorización en Twitter cada vez que se haga una petición de parte del usuario. Es decir, acompañarán a la petición y le harán saber a Twitter que este usuario nos concedió el permiso para hablar en su nombre.
5. La aplicación guardará las credenciales obtenidas para el usuario en las *SharedPreferences* de la aplicación, un sistema de guardado de pares clave-valor mediante XML que Android proporciona para guardar parámetros de uso de la aplicación sin recurrir a las bases de datos.
6. Si la aplicación posee valores para el "*Consumer secret*" y "*Consumer key*" del usuario en las *SharedPreferences*, se sobreentenderá que este usuario ya aceptó el uso de la aplicación y podrá ir a la pantalla de opciones. Si no se poseen dichos valores, solamente habrá acceso a la pantalla de autorización de la aplicación en Twitter. Por tanto, el proceso de "*logout*" de la aplicación consistirá en eliminar dichos valores de la aplicación, pero será en la web de Twitter donde el usuario deberá revocar realmente el acceso de la aplicación a su propia cuenta.

3.4.6.4 Datos devueltos por Twitter en las peticiones de búsqueda. Campos relevantes.

Cada vez que se hace una petición al servicio de búsqueda de tuits que ofrece Twitter, en este caso mediante la librería Twitter4j, se obtiene gran cantidad de datos en la respuesta que pueden consultarse en su documentación [URL54]. En nuestro caso se pedirán solamente los tuits originados a partir de la fecha de inicio del evento.

En pocas palabras, se ofrece una colección de tuits (que Twitter llama "statuses") con toda la información pública del tuit y del usuario que lo escribió. Puede verse un ejemplo de estatus devuelto en la tabla 61.

Para nuestra aplicación únicamente se hace uso de un conjunto muy reducido de la respuesta y esto es, los campos “created_at”, “id” y “text” de la sección relativa al tuit y el campo “screen_name” de la información perteneciente al usuario que originó el tuit.

Se guardará el identificador del último tuit que se muestra en el chat para poder solicitarle a Twitter solamente los tuits posteriores al que tiene dicho identificador.

```
1. "statuses": [  
2.   {  
3.     "coordinates": null,  
4.     "favorited": false,  
5.     "truncated": false,  
6.     "created_at": "Mon Sep 24 03:35:21 +0000 2012",  
7.     "id_str": "250075927172759552",  
8.     "entities": {  
9.       "urls": [  
10.      ],  
11.     },  
12.     "hashtags": [  
13.       {  
14.         "text": "freebandnames",  
15.         "indices": [  
16.           20,  
17.           34  
18.         ]  
19.       }  
20.     ],  
21.     "user_mentions": [  
22.     ]  
23.   },  
24.   "in_reply_to_user_id_str": null,  
25.   "contributors": null,  
26.   "text": "Aggressive Ponytail #freebandnames",  
27.   "metadata": {  
28.     "iso_language_code": "en",  
29.     "result_type": "recent"  
30.   },  
31.   "retweet_count": 0,  
32.   "in_reply_to_status_id_str": null,  
33.   "id": 250075927172759552,  
34.   "geo": null,  
35.   "retweeted": false,  
36.   "in_reply_to_user_id": null,  
37.   "place": null,  
38.   "user": {  
39.     "profile_sidebar_fill_color": "DDEEF6",  
40.     "profile_sidebar_border_color": "C0DEED",  
41.     "profile_background_tile": false,  
42.     "name": "Sean Cummings",  
43.     "profile_image_url":  
44.     "http://a0.twimg.com/profile_images/2359746665/1v6zfgqo8g0d3mk7ii5s_  
45.     normal.jpeg",  
46.     "created_at": "Mon Apr 26 06:01:55 +0000 2010",  
47.     "location": "LA, CA",  
48.     "follow_request_sent": null,
```

```

48.     "profile_link_color": "0084B4",
49.     "is_translator": false,
50.     "id_str": "137238150",
51.     "entities": {
52.         "url": {
53.             "urls": [
54.                 {
55.                     "expanded_url": null,
56.                     "url": "",
57.                     "indices": [
58.                         0,
59.                         0
60.                     ]
61.                 }
62.             ]
63.         },
64.         "description": {
65.             "urls": [
66.                 ]
67.             }
68.         },
69.         "default_profile": true,
70.         "contributors_enabled": false,
71.         "favourites_count": 0,
72.         "url": null,
73.         "profile_image_url_https":
74.         "https://si0.twimg.com/profile_images/2359746665/1v6zfgqo8g0d3mk7ii5
75.         s_normal.jpeg",
76.         "utc_offset": -28800,
77.         "id": 137238150,
78.         "profile_use_background_image": true,
79.         "listed_count": 2,
80.         "profile_text_color": "333333",
81.         "lang": "en",
82.         "followers_count": 70,
83.         "protected": false,
84.         "notifications": null,
85.         "profile_background_image_url_https":
86.         "https://si0.twimg.com/images/themes/theme1/bg.png",
87.         "profile_background_color": "CODEED",
88.         "verified": false,
89.         "geo_enabled": true,
90.         "time_zone": "Pacific Time (US & Canada)",
91.         "description": "Born 330 Live 310",
92.         "default_profile_image": false,
93.         "profile_background_image_url":
94.         "http://a0.twimg.com/images/themes/theme1/bg.png",
95.         "statuses_count": 579,
96.         "friends_count": 110,
97.         "following": null,
98.         "show_all_inline_media": false,
99.         "screen_name": "sean_cummings"
100.     },
101.     "in_reply_to_screen_name": null,
102.     "source": "<a
103.         href='\"http://itunes.apple.com/us/app/twitter/id409789998?mt=12\"'
104.         rel='\"nofollow\"'>Twitter for Mac</a>",
105.     "in_reply_to_status_id": null
106. },

```

Tabla 61. Ejemplo de información sobre un tuit que devuelve la API de Twitter

3.4.6.5 Obtención de código para el empotrado en la aplicación de los videos de youtube y presentaciones de Slideshare a partir de la URL proporcionada.

Otra característica de la aplicación es que los usuarios solamente proporcionarán la URL donde se aloja el material que quieren añadir al evento. Por tanto, es a partir de dicha URL de donde se deberá obtener (en el caso de YouTube y de Slideshare) el código HTML a empotrar en el componente WebView de Android, utilizado para mostrar el material de dichos tipos.

Esto lo haremos mediante el servicio *oEmbed* ofrecido por ambas aplicaciones en el momento de la creación del evento.

Un detalle importante es que dicha conversión entre la URL y el código a empotrar se realiza en el servidor para rebajar el consumo de datos del cliente, ya que es innecesario que sea la aplicación quien haga la petición, obtenga el texto y lo envíe al servidor consumiendo con ello datos que el usuario tal vez está pagando. De esta manera aliviamos el consumo, dejando al servidor hacer estas gestiones.

oEmbed

oEmbed (<http://oembed.com/>) es un servicio que ofrecen muchas aplicaciones mediante el cual podemos conseguir, de manera anónima, información pública y útil sobre uno de sus recursos. Normalmente le facilitaremos la URL del recurso sobre el que deseamos obtener información como un parámetro en la petición GET y nos devolverá la información como se ve en el ejemplo de youtube contenido en la tabla 62, donde se ha resaltado el campo que contiene la información que nos interesa para la aplicación, el HTML necesario para empotrar el video o la presentación en el WebView:

```
{ "author_name" : "schmoyoho",  
  "author_url" : "http://www.youtube.com/user/schmoyoho",  
  "height" : 344,  
  "html" : "<iframe width=\"459\" height=\"344\"  
src=\"http://www.youtube.com/embed/bDOYN-6gdRE?feature=oembed\"  
frameborder=\"0\" allowfullscreen></iframe>",  
  "provider_name" : "YouTube",  
  "provider_url" : "http://www.youtube.com/",  
  "thumbnail_height" : 360,  
  "thumbnail_url" : "http://i1.ytimg.com/vi/bDOYN-6gdRE/hqdefault.jpg",  
  "thumbnail_width" : 480,  
  "title" : "Auto-Tune the News #8: dragons. geese. Michael Vick. (ft. T-Pain)",  
  "type" : "video",  
  "version" : "1.0",  
  "width" : 459  
}
```

Tabla 62. Ejemplo de respuesta a una petición oEmbed en youtube

3.4.6.6 API creada para obtener información de los eventos y almacenarlos.

Como se ha comentado anteriormente, se ha elegido una arquitectura Cliente/Servidor, lo que implica que se ha de tener un servidor que responda a peticiones del cliente.

En la aplicación se guardarán los eventos creados en un servidor en Internet, en una Base de Datos muy sencilla en MySQL, a cuyos datos tenemos acceso vía una API pública muy sencilla, creada en PHP, que ofrece varios servicios básicos:

Verificar la existencia de un evento mediante el hashtag

URL de acceso	http://shoomuc.zz.mu/API/existeHashtag.php
Parámetros	<ul style="list-style-type: none"> hashtag: un texto que representa al hashtag del evento sin el carácter “#”
Método HTTP	GET
Ejemplo de respuesta positiva	<code>{"success": "1"}</code>
Ejemplo de respuesta negativa	<pre>{ "success": "0", "error": "Mensaje de error" }</pre>

Tabla 63. Resumen del servicio para verificar la existencia de un hashtag

Registrar un nuevo evento.

URL de acceso	http://shoomuc.zz.mu/API/addEvent.php
Parámetros	<ul style="list-style-type: none"> hashtag: un texto que representa al hashtag del evento sin el carácter “#” nombre: Nombre del evento f_inicio: Fecha de inicio del evento en formato aaaa-mm-dd f_fin: Fecha de fin del evento en formato aaaa-mm-dd tags: Etiquetas del evento separadas por comas descripcion: Descripción del evento creador: Screen name del usuario que ha creado el evento.
Método HTTP	POST
Ejemplo de respuesta positiva	<code>{"success": "1"}</code>
Ejemplo de respuesta negativa	<pre>{ "success": "0", "error": "Mensaje de error" }</pre>

Tabla 64. Resumen del servicio para registrar un nuevo evento

Registrar materiales multimedia a un evento.

URL de acceso	http://shoomuc.zz.mu/API/addMultimedia.php
Parámetros	<ul style="list-style-type: none"> materiales: Un array de objetos JSON, cada uno con este formato: <ul style="list-style-type: none"> hashtag: un texto que representa al hashtag del evento al que pertenece el material, sin el carácter “#” titulo: título del material tipo: uno de los siguientes valores: “imagen”, “slideshare”, “youtube” url: URL donde se aloja el material descripcion: Descripción del evento
Método HTTP	POST
Ejemplo de respuesta positiva	<code>{"success": "1"}</code>
Ejemplo de respuesta negativa	<pre>{ "success": "0", "error": "Mensaje de error" }</pre>

Tabla 65. Resumen del servicio para registrar un nuevo material en un evento

Obtener la información de un evento.

URL de acceso	http://shoomuc.zz.mu/API/getEvent.php
Parámetros	<ul style="list-style-type: none"> hashtag: un texto que representa al hashtag del evento, sin el carácter “#”
Método HTTP	GET
Ejemplo de respuesta positiva	<pre>{ "creador" : "miguelsaddress", "descripcion" : "descripcion del evento madrid", "fecha_fin" : "2999-12-31", "fecha_inicio" : "2013-07-08", "hashtag" : "madrid", "multimedia" : [{ "descripcion" : "cancion pohui del grupo moldavo Carlas dreams%# featuring INNA", "embed_code" : "<iframe width=\"100%\" height=\"100%\" src=\"http://www.youtube.com/embed/0FOxxI5lZZ8?feature=oembed\" frameborder=\"0\" allowfullscreen></iframe>", "hashtag" : "madrid", "id" : "23", "tipo" : "YouTube", "titulo" : "Carlas dreams - POHUI", "url" : "http://www.youtube.com/watch?v=0FOxxI5lZZ8" }, { "descripcion" : "imagen del palacio de</pre>

	<pre> comunicaciones en la tarde noche", "embed_code" : "", "hashtag" : "madrid", "id" : "24", "tipo" : "Imagen", "titulo" : "Imagen de madrid", "url" : "http://www.7pm.es/wp- content/uploads/2011/11/Que-ver-en-Madrid.jpg" }, { "descripcion" : "presentacion de slideshare de prueba", "embed_code" : "<iframe src=\"http://www.slideshare.net/slideshow/embed _code/16755036\" width=\"100%\" height=\"100%\" frameborder=\"0\" marginwidth=\"100%\" marginheight=\"100%\" scrolling=\"no\" style=\"border:1px solid #CCC;border-width:1px 1px 0;margin-bottom:5px\" allowfullscreen webkitallowfullscreen mozallowfullscreen> </iframe> <div style=\"margin-bottom:5px\"> Cómo gestionar las Notificaciones Electrónicas Obligatorias (NEOS). Emisión y gestión de las NEOS de diferentes Administraciones Públicas - AEAT (final) from Cámara de Comercio e Industria de Madrid </div>", "hashtag" : "madrid", "id" : "25", "tipo" : "SlideShare", "titulo" : "Camara de comercio", "url" : "http://www.slideshare.net/camaramadrid/jornada -camara-aeat-final-6" }], "nombre" : "Madrid Event", "num_de_multimedias" : 3, "success" : "1", "tags" : "madrid%# testing" } </pre>
Ejemplo de respuesta negativa	<pre> { "success": "0", "error": "Mensaje de error" } </pre>

Tabla 66. Resumen del servicio para obtener la información de un evento

3.4.6.7 Decisiones de diseño impuestas por terceros

La principal limitación a tener en cuenta para el diseño y que viene fuertemente impuesta por terceros, es el tiempo de refresco del chat del evento. A la fecha de este escrito, Twitter permite un máximo de 180 peticiones a cada función por cada 15 minutos, lo cual se traduce a un máximo de 1 petición cada 5 segundos de media. [<https://dev.twitter.com/docs/rate-limiting/1.1>]

Aún así se ha decidido hacer una petición de refresco cada 10 segundos, lo que implica que puede escribir a un ritmo constante de un tuit cada 10 segundos, lo cual parece mucho más que suficiente para las características de la aplicación.

3.5 Segunda Iteración

Tras la primera iteración, el equipo de desarrollo se ha reunido con el cliente, el cual ha quedado satisfecho de manera general con la aplicación entregada, pero tiene algunas sugerencias de mejora.

3.5.1 Metas Generales

El objetivo de esta iteración es el de llevar a cabo las correcciones solicitadas por el cliente, las cuales se enumeran a continuación:

1. En la pantalla para ver el evento, en el menú lateral, agrupar los materiales por su tipo.
2. En la pantalla para ver el evento, ha de aparecer el hashtag del evento en la barra superior, en lugar del título del material.
3. En la pantalla para ver el evento, en el menú lateral, se deben acompañar los materiales con una imagen que nos deje distinguir de qué tipo son.
4. En la pantalla para ver el evento, en el chat se deben acompañar los tuits con la imagen del perfil de Twitter del usuario.
5. En la pantalla para ver el evento, en el chat se debe eliminar la fecha y hacer que aparezcan solamente el nombre de usuario en Twitter del usuario que hizo el tuit y su nombre visible., además de la imagen y del texto del tuit.
6. En el proceso de creación del evento, a la hora de decidir si se debe tuitear la creación del mismo, se debe cambiar el *"ToggleButton"* por un *"CheckBox"*, marcado por defecto.
7. En el proceso de creación del evento, a la hora de decidir si se debe informar de la creación del mismo, no se debe obligar a que el usuario introduzca la "@" para el nombre de usuario.

3.5.2 Toma de Requisitos

3.5.2.1 Requisitos de software funcionales

No se añaden nuevos requisitos funcionales.

3.5.2.2 Requisitos de software no funcionales

Requisitos de comprobación

No se añaden nuevos requisitos de comprobación, pero se eliminan los siguientes requisitos existentes:

RSNFC100 - ELIMINADO			
Título	Formato de usuarios a notificar		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSNFC90		
Descripción	Los usuarios introducidos en el campo “notificar a” en la creación del evento estarán separados por comas, su primer carácter será la arroba “@” y no contendrán espacios interiores (por ejemplo, no es posible un usuario como: “user” o como “@un usuario” pero sí “@unusuario”). La longitud mínima de caracteres de cada usuario será de 2 caracteres y el máximo de 15.		
Fuente	Cliente, http://support.twitter.com/entries/14609		

Tabla 67: Requisito de Software no funcional de comprobación 100 eliminado

Este requisito no tiene requisitos dependientes, por lo que no hay que modificar más requisitos de comprobación.

Requisitos de restricción

No se añade ningún requisito de restricción.

Requisitos de interfaz de usuario

RSNFIdU70	
Título	Mostrar Hashtag del evento
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	Ninguna
Descripción	Durante la visualización de un evento, ha de aparecer el hashtag del evento en la barra superior, junto al logo.
Fuente	Cliente

Tabla 68: Requisito de Software no funcional de interfaz de usuario 70

RSNFIdU80	
Título	Materiales multimedia agrupados por tipo en menú lateral
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSNFIdU20
Descripción	En el menú lateral para la selección de material multimedia que aparece durante la visualización de un evento, los materiales se han de agrupar por su tipo.
Fuente	Cliente

Tabla 69: Requisito de Software no funcional de interfaz de usuario 80

RSNFIdU90	
Título	Acompañar material de imagen representativa
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No Verificable <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSNFIdU20
Descripción	En el menú lateral para la selección de material multimedia que aparece durante la visualización de un evento, cada material del menú ha de acompañarse con una imagen que nos ayude a distinguir el tipo de material de que se trata.
Fuente	Cliente

Tabla 70: Requisito de Software no funcional de interfaz de usuario 90

RSNFIdU100			
Título	Imagen de usuario en el chat		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF120		
Descripción	En la pantalla de visualización de evento, en el chat, se ha de mostrar en cada entrada la imagen del perfil del usuario en Twitter para ajustarse más a los requerimientos de visualización de tuits de Twitter.		
Fuente	Cliente, https://dev.twitter.com/terms/display-requirements		

Tabla 71: Requisito de Software no funcional de interfaz de usuario 100

RSNFIdU110			
Título	Información de usuario a mostrar en el chat		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF120		
Descripción	En la pantalla de visualización de evento, en el chat, se ha de mostrar en cada entrada el nombre de usuario en Twitter del usuario que generó el tuit, junto con su nombre público (campos <i>screen_name</i> y <i>name</i> de la respuesta de Twitter) para ajustarse más a los requerimientos de visualización de tuits de Twitter..		
Fuente	Cliente, https://dev.twitter.com/terms/display-requirements		

Tabla 72: Requisito de Software no funcional de interfaz de usuario 110

RSNFIdU110			
Título	Checkbox para decidir si informar de creación de evento		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Estabilidad	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No	Verificable	<input checked="" type="checkbox"/> Sí <input type="checkbox"/> No
Dependencias	RSF50		
Descripción	Durante el proceso de creación de un nuevo evento, se ha de poder decidir si se ha de emitir un tuit comunicando la creación del evento mediante el uso de un CheckBox en lugar de un ToggleButton, que ha de estar marcado por defecto.		
Fuente	Cliente		

Tabla 73: Requisito de Software no funcional de interfaz de usuario 110

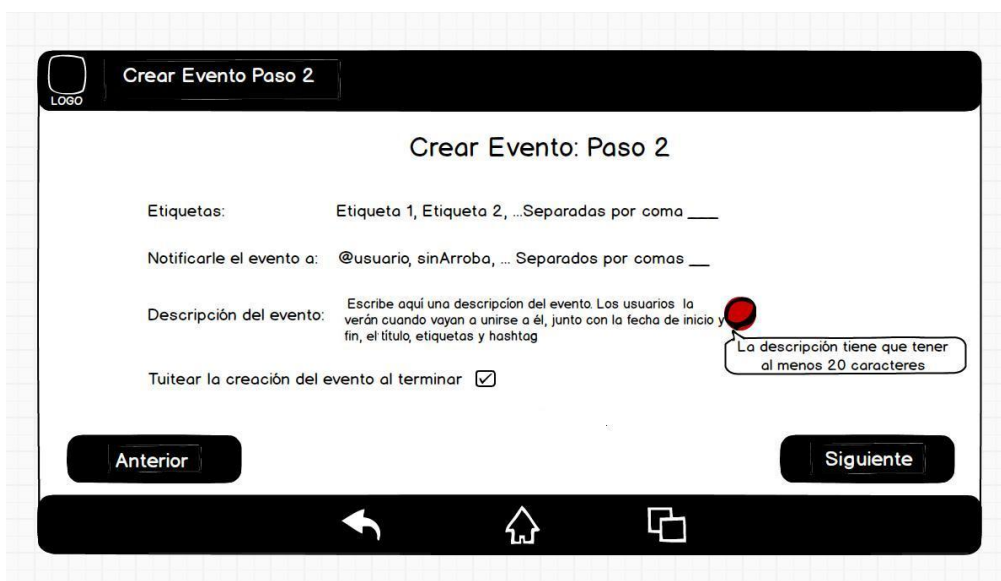
3.5.3 Casos de uso de la aplicación

No se añaden nuevos casos de uso en la aplicación.

3.5.4 Wireframes

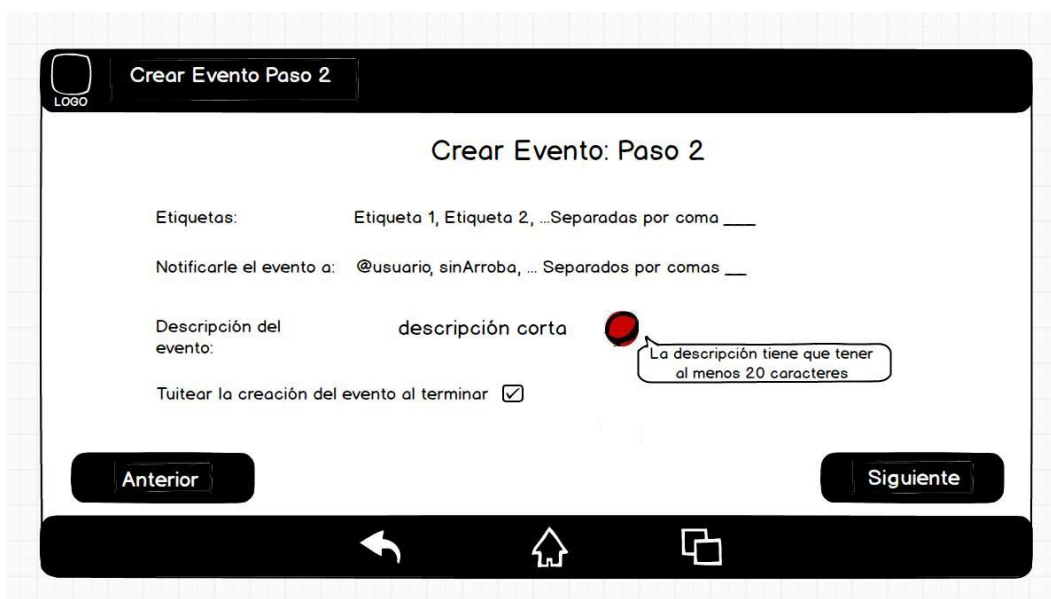
No se añaden nuevas pantallas en la aplicación y los cambios requeridos por el cliente no modifican la funcionalidad, por lo que la mayoría de los wireframes actuales emitidos durante la iteración 1 siguen siendo válidos, a excepción de los que forman parte del paso 2 de la creación de evento.

Ya no se debe notificar error al introducir usuarios sin la arroba como primer carácter y el “ToggleButton” pasa a ser un “CheckBox”, siendo el resultado de los dos wireframes modificados el siguiente:



The wireframe shows a mobile app screen titled "Crear Evento Paso 2". It has a header bar with a logo and the title. The main content area contains several form fields: "Etiquetas:" with a placeholder "Etiqueta 1, Etiqueta 2, ...Separadas por coma ____", "Notificarle el evento a:" with a placeholder "@usuario, sinArroba, ... Separados por comas ____", and "Descripción del evento:" with a placeholder "Escribe aquí una descripción del evento. Los usuarios la verán cuando vayan a unirse a él, junto con la fecha de inicio y fin, el título, etiquetas y hashtag". Below these fields is a checkbox labeled "Tuitear la creación del evento al terminar" which is checked. At the bottom are two buttons: "Anterior" and "Siguiete". A red circle highlights the "Descripción del evento:" field, and a speech bubble points to it with the text "La descripción tiene que tener al menos 20 caracteres".

Figura 49. Boceto de error en el segundo paso de la creación de evento reeditado, descripción vacía



The wireframe shows the same mobile app screen as Figure 49, but with a different error message. The "Descripción del evento:" field now contains the text "descripción corta". A red circle highlights this text, and a speech bubble points to it with the text "La descripción tiene que tener al menos 20 caracteres". The other elements of the screen, including the header, form fields, checkbox, and buttons, remain the same.

Figura 50. Boceto de error en el segundo paso de la creación de evento reeditado, descripción demasiado corta

3.5.5 Implementación

En esta iteración se ha implementado una caché de imágenes, motivada por dos factores principalmente:

El primero de ellos es disminuir en la medida de lo posible el consumo de datos, ya que cada vez que un usuario decidía visualizar una de las imágenes, ésta se descargaba de Internet y se mostraba en un elemento “ImageView”.

El segundo y el que motivó el estudio de la implementación de una caché fue la inclusión de las imágenes de perfil de usuario en Twitter en cada elemento del chat. Esto presentaba un extraño comportamiento, ya que las imágenes se han de descargar en un hilo secundario que no sea el de la interfaz de usuario y aunque se tratara de imágenes de pequeño tamaño, se producía un retardo en la carga y con cada refresco del chat se debían obtener nuevamente, provocando que se mostrara el texto en los tuits del chat pero las imágenes aparecieran con un “efecto ola” o de refresco continuo.

Para la implementación de la caché se ha seguido la pauta marcada por el manual oficial de desarrollo para Android [URL55]. Para ello se ha creado una clase envoltorio de la caché, llamada “ImagesCache” que ha de combinar su uso con la clase encargada de descargar imágenes de Internet y el uso es muy sencillo:

Cuando se requiere una imagen alojada en Internet, ya sea como material multimedia o como imagen de usuario en el chat, primero se busca en la caché, donde se referencia mediante la URL que la aloja. Si se encuentra en la caché, se obtiene y se muestra, y si no se encuentra, se ejecuta la descarga habitual de imágenes. Una vez descargada la imagen, se almacena en la caché, utilizando como clave la URL de donde se obtuvo.

3.6 Resultado Final de la aplicación

A continuación se añaden unas capturas de pantalla con el resultado final de la aplicación.

3.6.1 Pantalla inicial de la aplicación

Es la pantalla donde el usuario tiene la opción de comenzar el proceso de autenticación en Twitter y obtener las credenciales para que la aplicación pueda comunicarse con Twitter en su nombre.



Figura 51. Pantalla inicial de la aplicación

3.6.2 Pantalla de opciones

Una vez que el usuario está autenticado en la aplicación, la pantalla principal de la aplicación permite elegir si quiere crear un nuevo evento o si quiere unirse a uno existente.

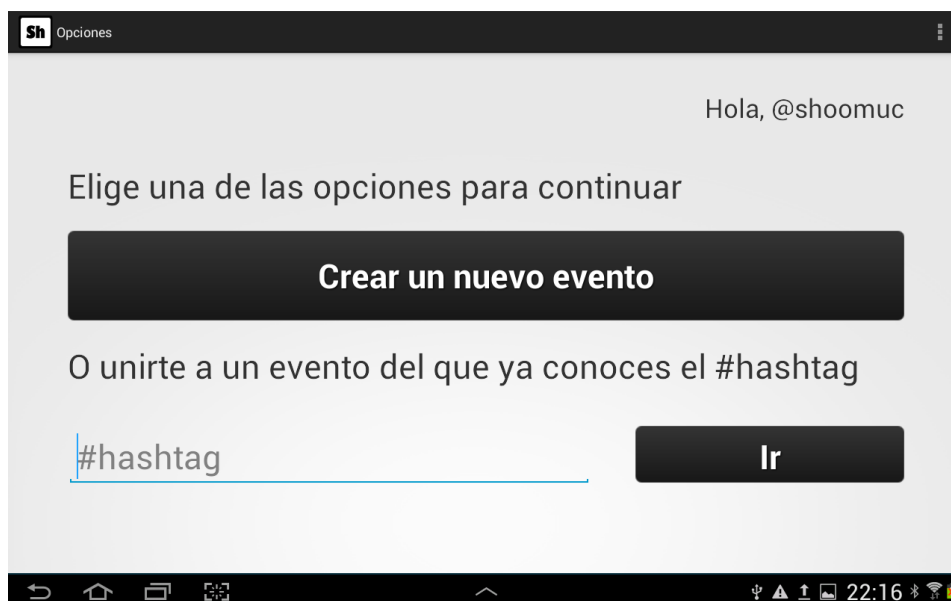


Figura 52. Pantalla de opciones

A partir de este punto se distinguen 2 escenarios: Creación de evento y visualización de un evento.

3.6.3 Escenario A: Creación de evento

Para crear un evento se ha de seguir un recorrido de 3 pasos.

3.6.3.1 Primer paso de creación de evento

El usuario debe introducir la información básica del evento: Un nombre, un hashtag y una fecha de inicio del mismo.

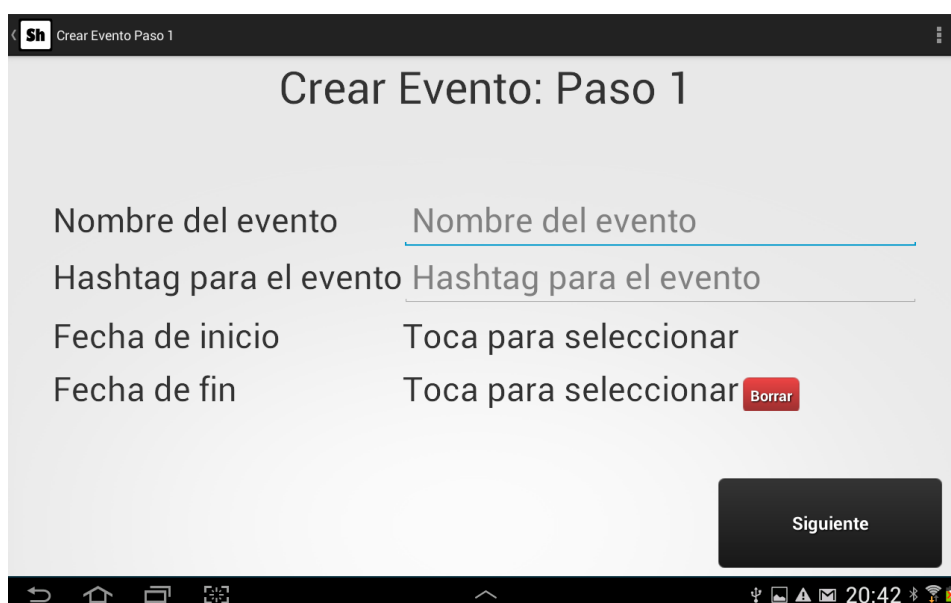


Figura 53. Paso 1 de la creación de un evento

En la siguiente captura vemos que el teclado virtual de Android no solapa el contenido, dejando ver los textos que se están introduciendo.

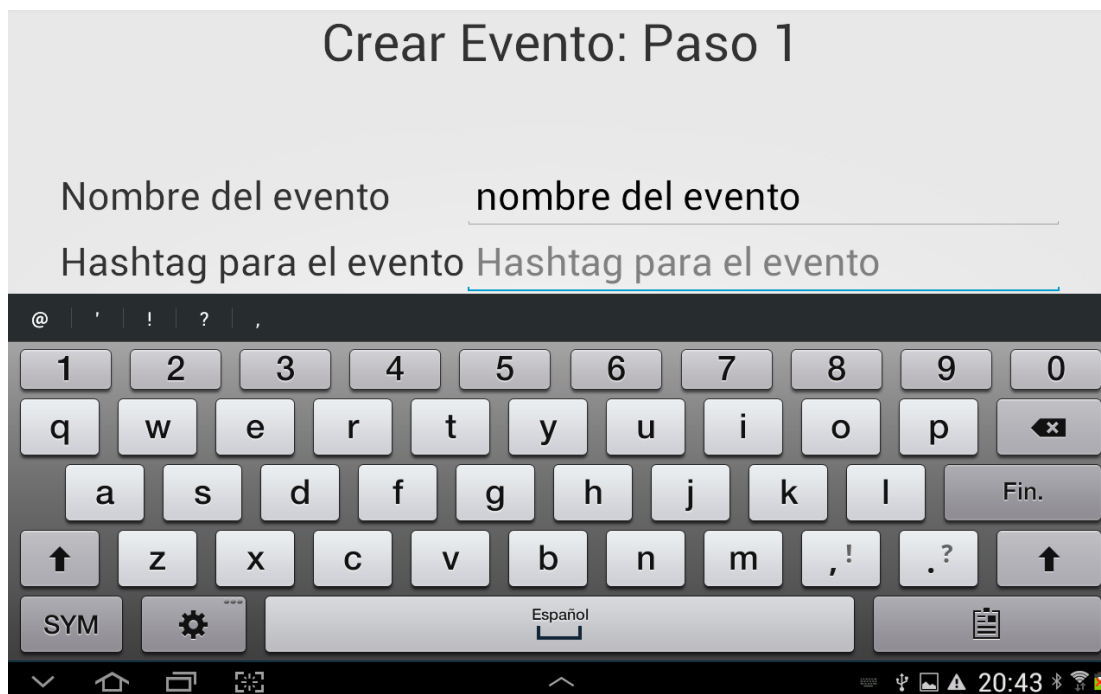


Figura 54. Teclado virtual de Android sobre el paso 1

En la próxima captura, vemos el calendario que nos ofrece Android para elegir las fechas del evento

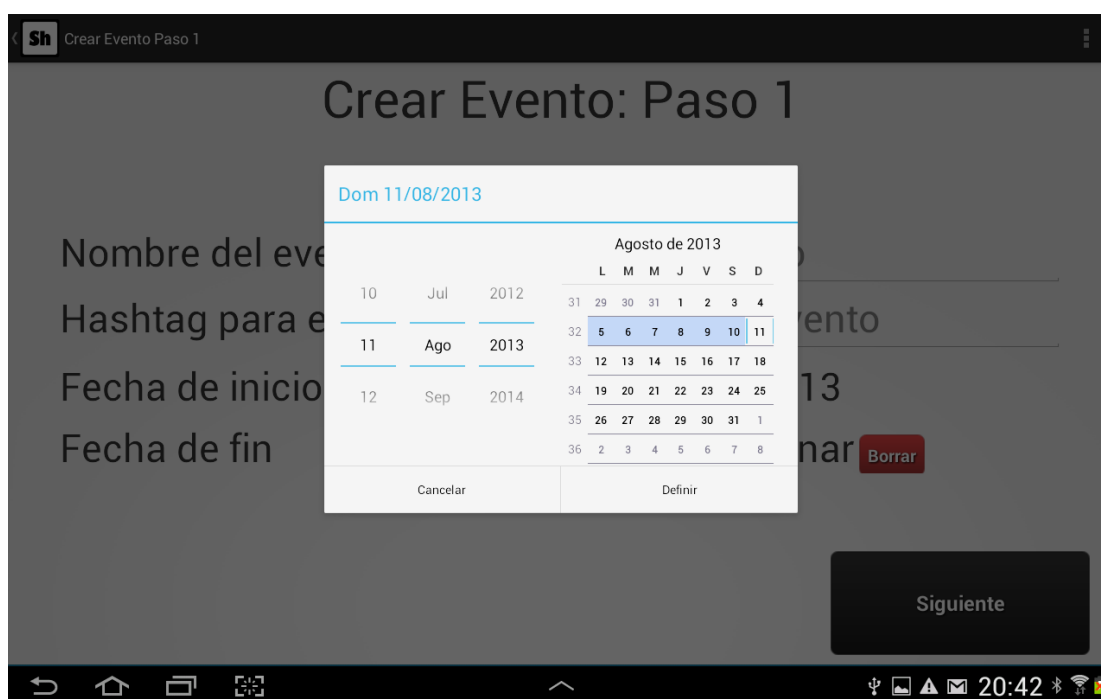


Figura 55. Calendario para la introducción de fechas

El resultado final del primer paso será algo como sigue a continuación:



Figura 56. Primer paso debidamente cumplimentado

3.6.3.2 Segundo paso de creación de evento

En este segundo paso se han de introducir datos informativos el evento, como las etiquetas, su descripción o decidir si le damos publicidad mediante el uso de Twitter.



Figura 57. Segundo paso de creación de evento sin completar

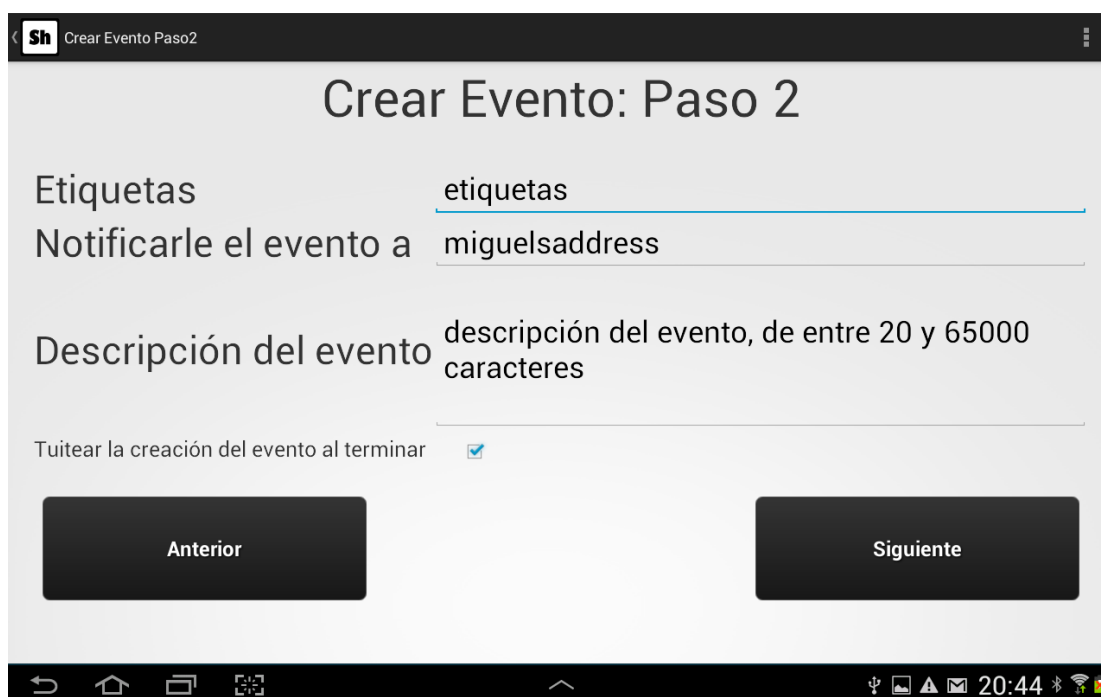


Figura 58. Segundo paso de creación de evento completado

3.6.3.3 Tercer paso de la creación de evento

En el último paso es donde el usuario debe añadir los materiales multimedia uno a uno, eligiendo primero el tipo de material a añadir.

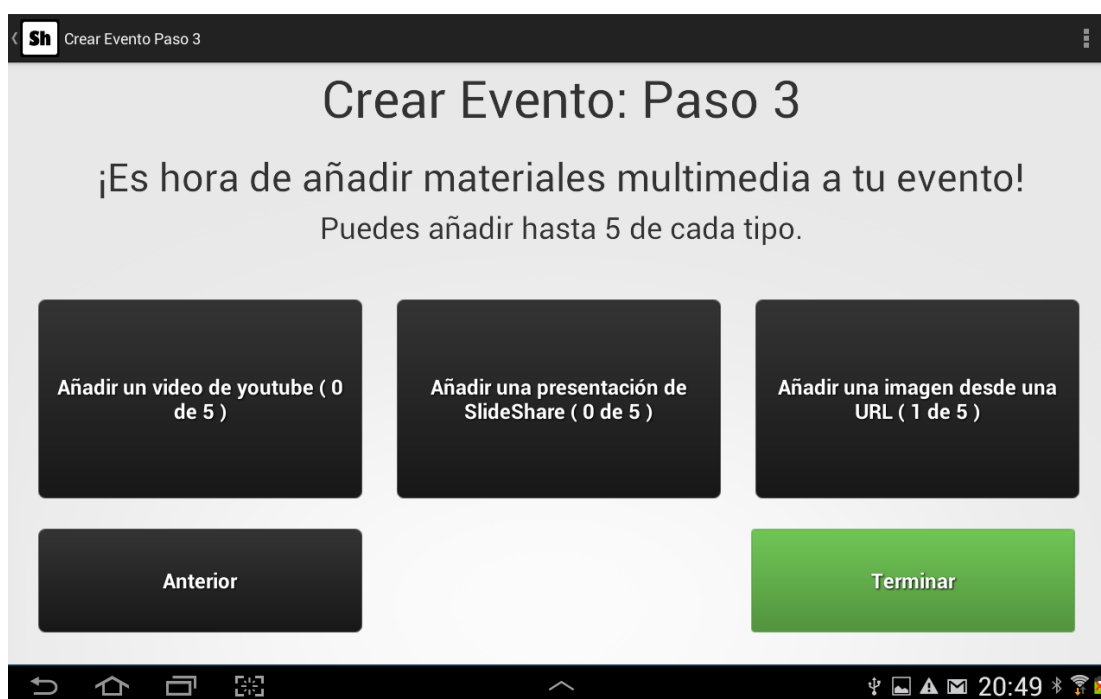


Figura 59. Tercer paso de la creación de evento

Si el usuario no añade ningún material, recibirá una notificación del error.

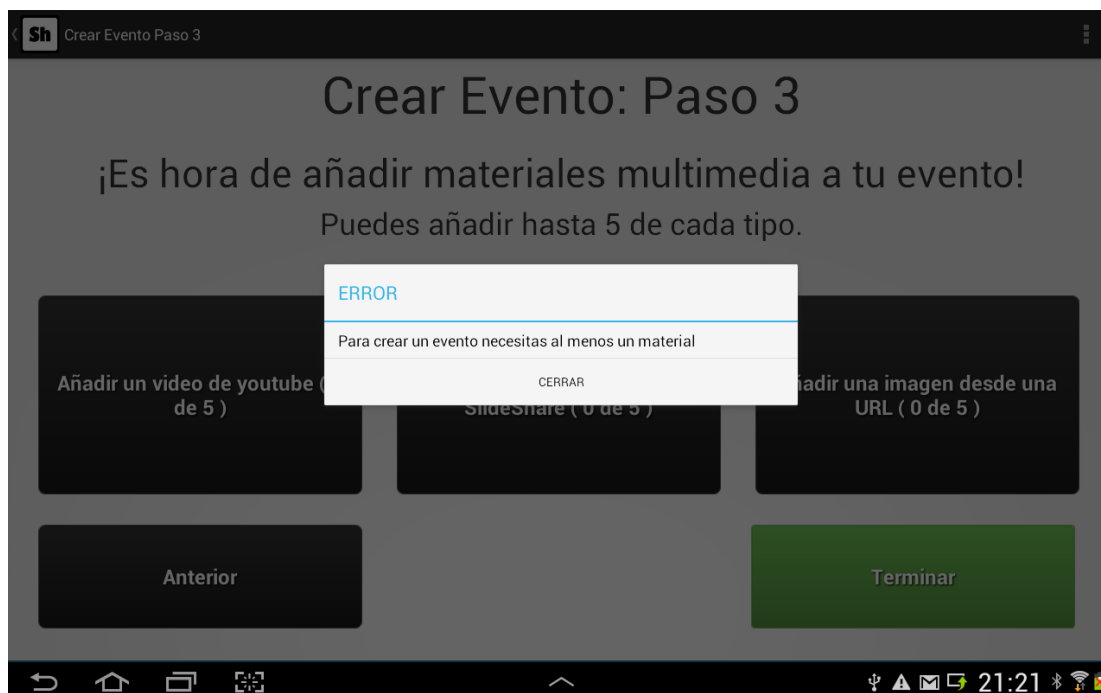


Figura 60. Notificación de error mediante ventana emergente

Cuando el usuario selecciona un tipo de material a añadir, por ejemplo una imagen, deberá introducir para cada material un título, una descripción y proporcionar la URL donde se aloja el material.



Figura 61. Pantalla para añadir un nuevo material multimedia

Si algún campo no fuera correcto, se le notificará al usuario el error:



The screenshot shows a mobile application interface for adding new multimedia material. The title is 'Añadir Nuevo Material Multimedia'. Below it, the category 'Imagen' is selected. There are three input fields: 'Título' with the value 'una imagen para el evento', 'URL' with the placeholder 'URL donde está el material', and 'Descripción' with the placeholder 'descripción del material, entre 20 y 500 caracteres'. A red box highlights the URL field, and a red error message bubble appears next to it, stating 'La URL debe tener entre 10 y 500 caracteres'. At the bottom, there are two buttons: 'Cancelar' and 'Aceptar'. The Android status bar at the bottom shows the time as 21:24.

Figura 62. Error durante la creación de un nuevo material para un evento

3.6.4 Escenario B: Visualización de un evento

Cuando un usuario introduce un hashtag en la pantalla de opciones y pulsa sobre el botón "Ir", accede al evento creado mediante la aplicación con dicho hashtag. Si el hashtag no existe en la aplicación, se le notificará oportunamente.

Si el hashtag existe, el usuario es conducido a una pantalla donde visualiza la información general del evento, pudiendo decidir si abrir el mismo o volver a la pantalla de opciones. Esta pantalla es la que muestra la figura 64.

Si el usuario ha decidido visualizar el evento, se le conduce a la pantalla de visualización del mismo, con el primer elemento multimedia activo.

En la figura 65 podemos ver cómo se muestra una imagen en la pantalla, en la figura 66 podemos ver la visualización de un video de youtube y en la figura 67 podemos ver la visualización de un material Slideshare.

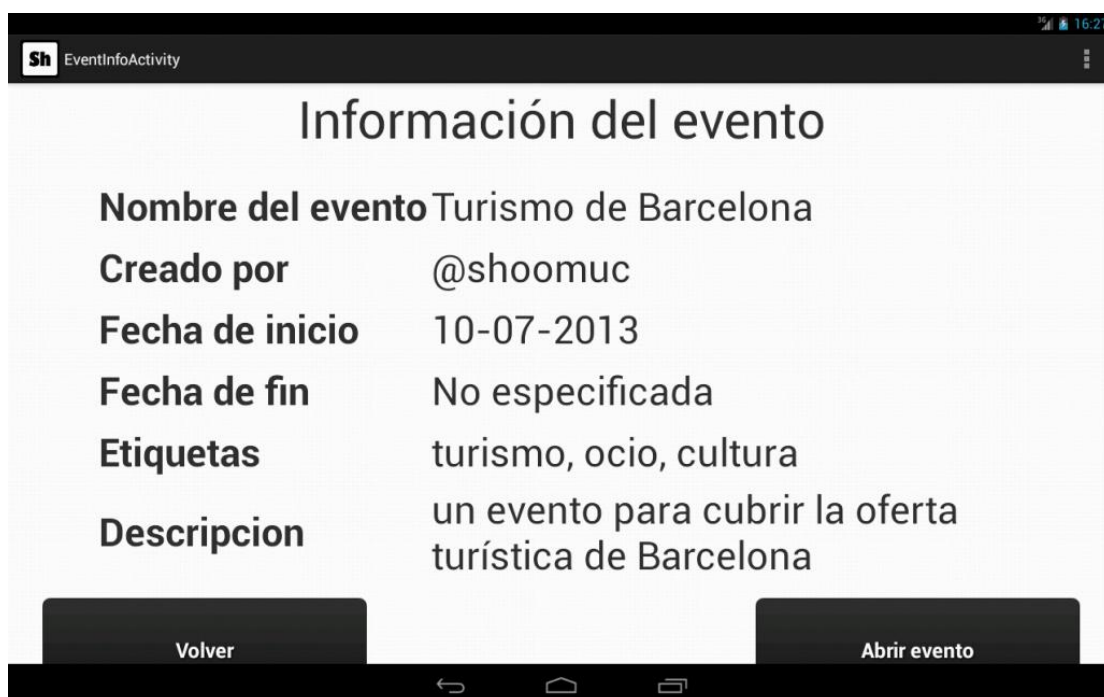


Figura 63. Datos generales del evento

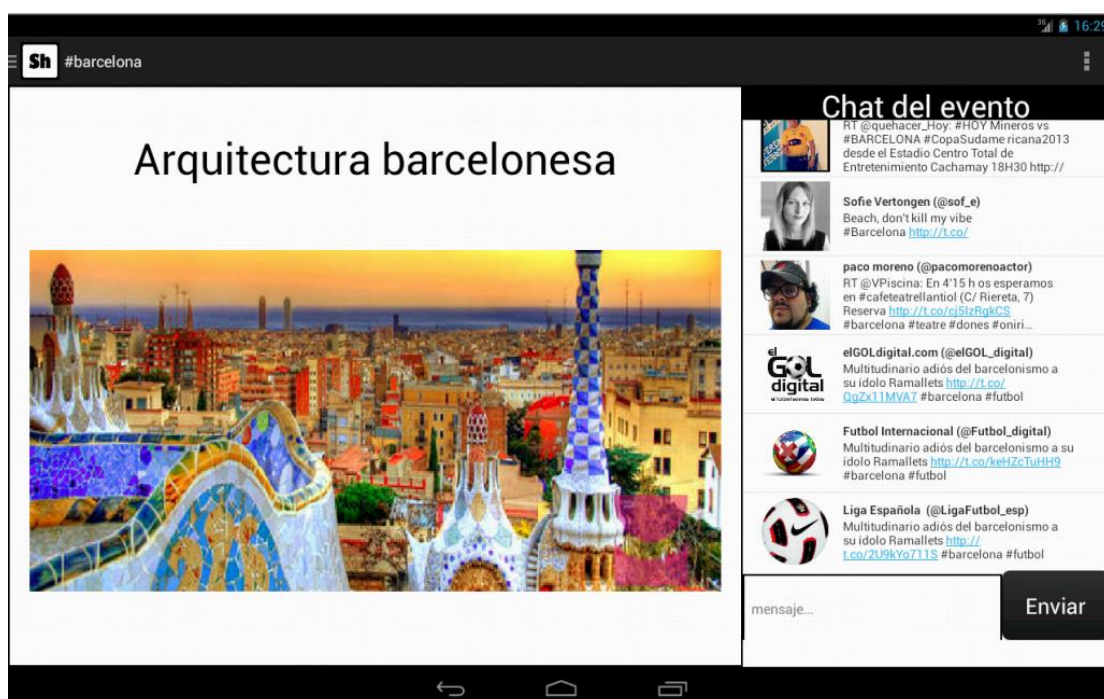


Figura 64. Visualización de una imagen



Figura 65. Visualización de un video de Youtube



Figura 66. Visualización de una presentación Slideshare

Si el usuario decide abrir el menú lateral, el cual se acciona pulsando el logo de la aplicación, este solapa el material que esté siendo visualizado en una lista de elementos agrupados por su tipo, indicando a través de un icono el tipo de elemento que se trata: una cámara para las imágenes, el logo de Youtube para los videos de dicha plataforma y el logo de Slideshare para los pases de diapositivas.

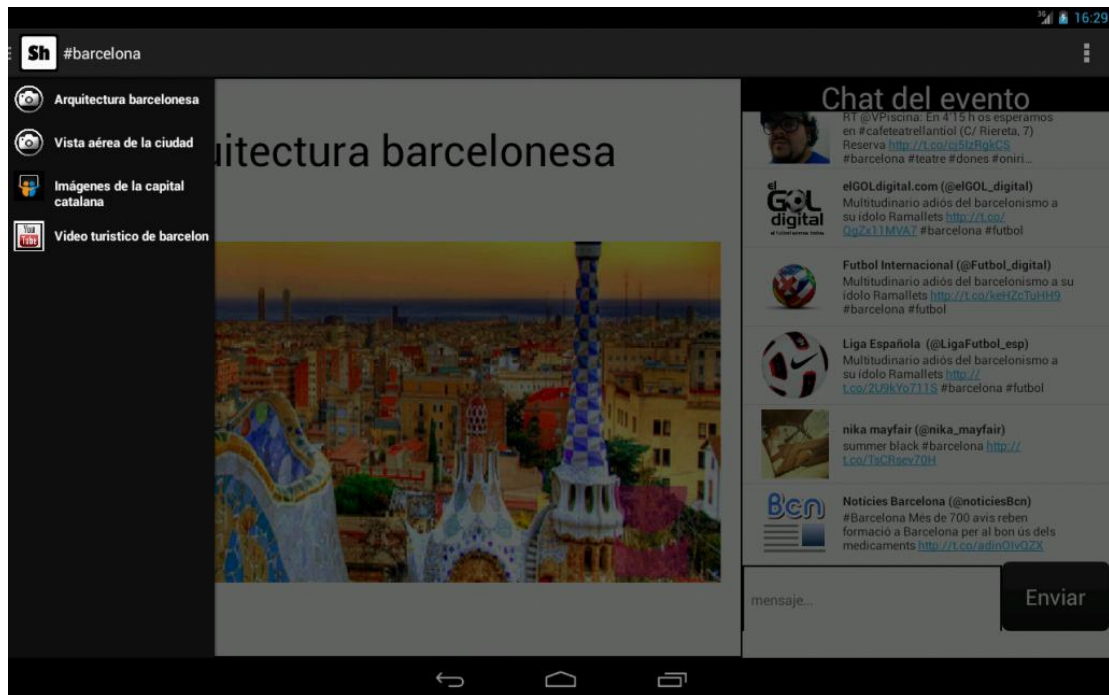


Figura 67. El menú deslizante de la aplicación activo

4. Planificación

En este capítulo se incluye una planificación detallada del proyecto, indicando las fechas de inicio y de fin de cada fase, el esfuerzo dedicado a cada una de ellas y una representación gráfica mediante un diagrama de Gantt.

Para la contabilización de los días empleados en el proyecto se tiene en cuenta una jornada laboral de 5 horas diarias de lunes a viernes.

El desarrollo del proyecto se ha dividido en 3 fases principales:

1. **Investigación y estudio de las alternativas:** Esta fase engloba tanto la definición del problema donde el cliente nos indica su necesidad y evaluamos las posibles soluciones, como el estudio de las aplicaciones y tecnologías existentes a la fecha de inicio del proyecto. La fase concluye seleccionando una de las tecnologías propuestas.
2. **Desarrollo de la aplicación:** En esta segunda fase se ha llevado a cabo la implementación de la solución siguiendo una metodología de desarrollo de tipo ágil, por lo que el desarrollo ha quedado dividido en iteraciones. Cada iteración se compone principalmente de las siguientes subfases:
 - a. Identificación de los requisitos
 - b. Wireframes
 - c. Desarrollo o Implementación
 - d. Documentación asociada
3. **Documentación final:** Generación del documento entregable final, revisiones, correcciones, estructuración y preparación de la presentación.

Esta planificación queda resumida en la siguiente imagen:

Tarea	Esfuerzo	Inicio	Fin
1) Investigación y estudio de alternativas	8s 3d	18/03/13 08:00	15/05/13 17:00
2) Desarrollo de la aplicación	11s	16/05/13...	26/07/13...
2.1) Iteración 1	8s 4d	16/05/13...	12/07/13...
2.1.1) Identificación de requisitos	1s 2d	16/05/13 08:00	24/05/13 17:00
2.1.2) Wireframes	1s	27/05/13 08:00	31/05/13 17:00
2.1.3) Desarrollo	5s 1d	03/06/13 08:00	08/07/13 17:00
2.1.4) Documentación	1s 1d	05/07/13 08:00	12/07/13 17:00
2.2) Iteración 2	2s 1d	15/07/13...	26/07/13...
2.2.1) Identificación de requisitos	1d	15/07/13 08:00	15/07/13 17:00
2.2.2) Wireframes	1d	15/07/13 08:00	15/07/13 17:00
2.2.3) Desarrollo	1s 2d	16/07/13 08:00	24/07/13 17:00
2.2.4) Documentación	2d	25/07/13 08:00	26/07/13 17:00
3) Documentación final	5s	29/07/13 08:00	30/08/13 17:00

Figura 68. Resumen de la planificación mostrando fechas de inicio y fin de cada tarea y su esfuerzo

En la siguiente página puede verse el diagrama de Gantt que describe la planificación de manera gráfica.

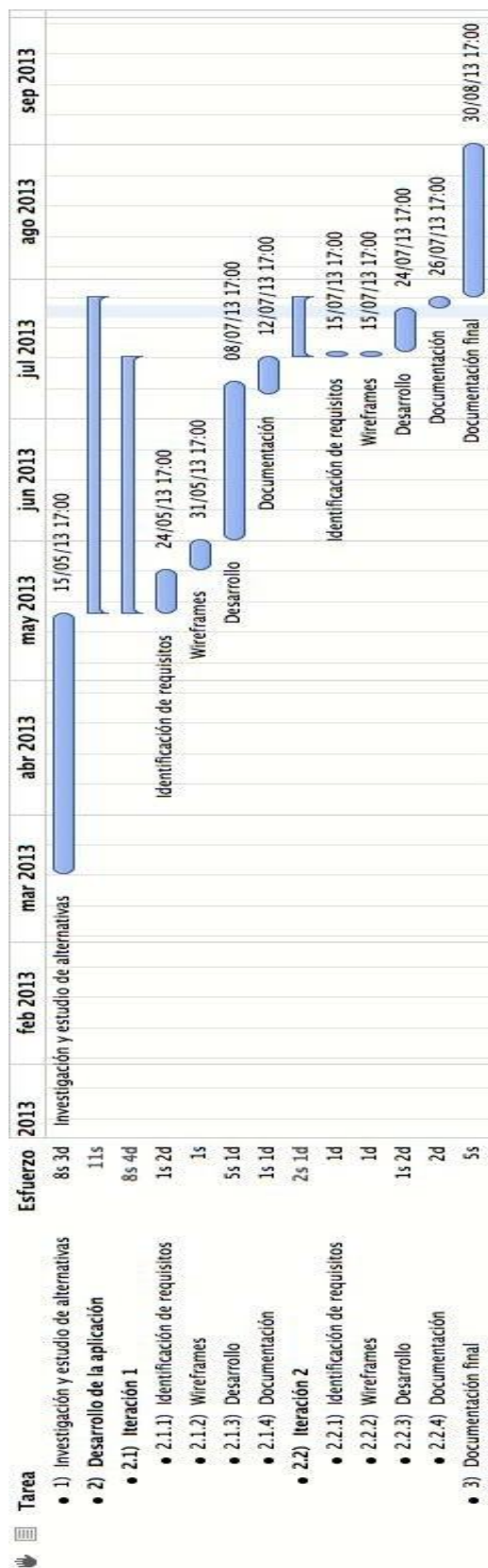


Figura 69. Diagrama de Gantt representativo de la planificación del proyecto

5. Presupuesto

En este apartado del documento se va a recoger el cálculo del presupuesto total del desarrollo de la aplicación. Para ello se han dividido los costes en dos tipos:

1. **Costes de personal:** El coste del personal será el resultante de multiplicar el total de horas dedicadas al proyecto por sus integrantes (en este caso solamente uno) por su coste por día.
2. **Costes de materiales:** Se tendrán en cuenta los costes tanto de equipos informáticos como los costes de software, aplicando solamente una parte de su valor cuando corresponda la amortización.

5.1 Costes de personal

Para poder evaluar los costes de personal se ha debido calcular primero una estimación del total de horas dedicadas por hombre y día al mismo. En nuestro caso tomaremos el tiempo en **días laborables** transcurrido de la planificación, un total de 120 días laborables.

La jornada diaria de trabajo estimada será de 5 horas diarias, a razón de 12 euros la hora de trabajo, es decir, 60 euros diarios.

Fase	Fecha de inicio	Fecha de fin	Días	Coste (€)
Investigación y estudio de las alternativas	18/03/2013	15/05/2013	43	43*60= 2580
Desarrollo de la aplicación	16/05/2013	26/07/2013	52	52*60= 3120
Generación de la documentación final	29/07/2013	30/08/2013	25	25*60= 1500
TOTAL			120	7200

Tabla 74. Resumen de costes de personal del proyecto

Por tanto, el coste de personal del proyecto es de 7.200 euros (SIETE MIL DOSCIENTOS EUROS).

5.2 Costes de material

Respecto al coste de material se va a distinguir entre coste software y coste hardware y se va a tener en cuenta un tiempo de amortización de 60 meses, excepto en el caso de la tableta de pruebas, comprada específicamente para el proyecto.

Debido a que el proyecto se inicia el 18 de marzo de 2013 y se da por finalizado el 30 de agosto de 2013, se tendrá en cuenta un periodo de 6 meses para calcular el coste aplicable del hardware o del software.

5.2.1 Costes de hardware

Dispositivo	Cantidad	Coste (€)	Coste mensual aplicable (€)	Coste total para 6 meses (€)
Samsung Galaxy Tab 10.1" 16GB	1	265	265 *	265
Apple MacBook Pro mid 2009 13.3"	1	920,4	15,34	92,04
TOTAL				357,04

Tabla 75. Costes de hardware del proyecto

El coste del hardware empleado es de 357,04 euros (TRESCIENTOS CINCUENTA Y SIETE EUROS CON CUATRO CÉNTIMOS).

5.2.2 Costes de software

Programa	Coste (€)	Coste mensual aplicable (€)	Coste total para 6 meses (€)
Office for Mac	119	1,98	11,90
Eclipse	Gratuito	0	0
Hosting	Gratuito	0	0
Cyberduck	Gratuito	0	0
Git	Gratuito	0	0
BitBucket	Gratuito	0	0
OmniGraffle	89,99	1,50	9,00
OmniPlan	179,99	3,00	18,00
TOTAL			38,90

Tabla 76. Costes de software del proyecto

El coste total del software atribuible al proyecto es de 38,90 euros (TREINTA Y OCHO EUROS CON NOVENTA CÉNTIMOS).

* El coste del dispositivo Android se carga íntegro al proyecto.

5.2.3 Coste total del material

Concepto	Coste
Costes de hardware	357,04 €
Costes de software	38,90 €
TOTAL	395,94 €

Tabla 77. Resumen de gastos de material atribuibles al proyecto

Por tanto el coste total de materiales es de 395,94 euros (TRESCIENTOS NOVENTA Y CINCO EUROS CON NOVENTA Y CUATRO CÉNTIMOS).

5.3 Coste total del proyecto

Una vez que se han calculado los gastos materiales y los gastos de personal, se puede calcular el coste total del proyecto, al que añadiremos un 10% de gastos indirectos (transporte, electricidad...).

Concepto	Coste
Costes de personal	7.200 €
Costes de hardware	395, 94 €
SUBTOTAL	7.595,94 €
10% de costes indirectos	759,59
TOTAL	8.355,53

Tabla 78. Coste total del proyecto sin 21% IVA

El coste total del proyecto sin IVA es de 8.355,53 euros (OCHO MIL TRESCIENTOS CINCUENTA Y CINCO EUROS CON CINCUENTA Y TRES CÉNTIMOS).

El coste total del proyecto con el 21% de IVA es de 10.110,19 euros (DIEZ MIL CIENTO DIEZ EUROS CON DIECINUEVE CÉNTIMOS).

6. Conclusiones y líneas futuras

6.1 Conclusiones

En esta sección se analiza la solución desarrollada con el fin de comprobar que cumple los objetivos identificados. Para ello se utilizará una tabla como la utilizada en el apartado “1.2.7 Comparación de las aplicaciones existentes”, donde se analizaban aplicaciones existentes previamente al desarrollo de la aplicación de este proyecto.

En la siguiente tabla, se muestran las características utilizadas en dicha comparativa y el soporte proporcionado por Shoomuc, a modo de evaluación de la solución propuesta.

Característica	Shoomuc
Chat entre asistentes	Sí, a través de Twitter
Visualización del multimedia Online	Sí
Tipos de multimedia ofrecidos	Enlaces, imágenes, pases de diapositivas de SlideShare, videos de YouTube
Tiene integración con Redes Sociales	Twitter
Participantes simultáneos	Tantos como sea necesario
Dispone de versión para dispositivos móviles	Sí, Android

Tabla 79. Evaluación de la aplicación desarrollada

Una vez presentado el diseño y desarrollo de la aplicación, se puede observar que se ha cumplido el objetivo principal de este proyecto: proporcionar una herramienta que permita compartir opiniones y comentarios sobre diferentes tipos de materiales multimedia mediante el uso de las redes sociales.

Esto se ha llevado a cabo mediante el desarrollo de una aplicación para la plataforma Android que permite la visualización de elementos multimedia de tipos representativos - como son videos, imágenes o pases de diapositivas - así como conversar sobre ellos mediante el uso de las redes sociales, en este caso, Twitter.

Los factores evaluados son los siguientes:

Chat entre los asistentes

Gracias a la integración con la red social twitter y el uso de un hashtag para el evento, es posible crear un chat mediante la recuperación de aquellos mensajes que lo utilizan y publicando nuestros mensajes con dicho hashtag.

Visualización de multimedia online

Los materiales multimedia permitidos son tomados de la web y pueden visionarse siempre que se tenga conexión a internet para acceder a ellos.

Tipos de multimedia ofrecidos

Se ofrecen materiales de tipos representativos, como son las imágenes, los videos (utilizando YouTube) y los pases de diapositivas (mediante Slideshare).

Integración con redes sociales

Se utiliza la red social Twitter tanto para acceder a la aplicación como para interactuar en el chat, haciendo así que el usuario no tenga que registrarse en el sistema Shoomuc y pueda utilizar su cuenta de Twitter.

Participantes simultáneos

Puede haber tantos participantes como usuarios envíen mensajes con el hashtag utilizado.

Dispone de versión para móvil

Se dispone de aplicación para tabletas de 7 y 10 pulgadas Android, la desarrollada en este proyecto.

Se puede apreciar por tanto que la aplicación desarrollada proporciona las funcionalidades que se le requerían a las demás aplicaciones analizadas en el Estado del Arte, ya que, aunque en la versión para dispositivos móviles solamente se cuenta con versión para la plataforma Android, esta es una restricción fijada al inicio del desarrollo del proyecto con el fin de acotarlo.

6.2 Líneas futuras

Este apartado recoge una serie de ideas de mejora que la aplicación podría incorporar en el caso de continuar su desarrollo, bien con siguientes iteraciones o bien por parte de otro desarrollador que quiera continuar con ella.

Realizar una implementación para otras plataformas

Para poder ofrecer acceso a todo tipo de usuarios sería interesante la implementación de la aplicación tanto para dispositivos iOS como en su versión Web.

Edición y visualización de eventos creados por el usuario

Actualmente solo se pueden crear eventos pero no puede verse qué eventos ha creado el usuario ni tampoco editarlos, ya sea para corregir textos o para actualizar el listado de materiales multimedia.

Para ello, puede utilizarse el campo de la base de datos que indica qué usuario creó el evento, ya que se creó para esta finalidad, y a partir de ahí, obtener los eventos creados por el usuario actual y crear el listado y las actividades de edición de eventos.

Búsqueda de eventos

Una funcionalidad útil puede ser permitir la búsqueda de eventos a través de la aplicación. Con este fin, durante la creación de eventos se incluyen campos que pueden facilitar su búsqueda posterior, como las etiquetas y las fechas de inicio y fin del evento, que pueden utilizarse para buscar eventos que en su título, hashtag y descripción contengan cierto texto introducido por el usuario y filtrarlo por fechas y etiquetas.

Incorporar nuevos tipos de materiales

Durante la definición del alcance del proyecto se decidió que se incorporarían los materiales multimedia soportados actualmente: imágenes, videos alojados en youtube y presentaciones de Slideshare. Aunque es un espectro representativo de los diferentes materiales multimedia de uso cotidiano, podría ampliarse ofreciendo, por ejemplo, documentos alojados en Scribd.com o pudiendo crear una sección para añadir enlaces de interés del evento. También aumentar los proveedores de video, añadiendo por ejemplo Vimeo, Muzu.tv, etc. Esta mejora implicaría cambios transcendentales en la interfaz de usuario, modificando el tercer paso del proceso de creación de eventos y la interfaz de creación de nuevo material. Además habría que identificar dicho tipo de material en el servidor para que los scripts de creación de nuevos eventos y materiales los aceptaran.

Mayor enfoque a eventos con ponente

En la actualidad hay múltiples eventos que se apoyan en Twitter durante sus conferencias, permitiendo mandar mensajes con un cierto hashtag que se recogen como dudas a resolver por el ponente. Estaría bien añadir esta funcionalidad, de manera que el usuario tuviera algo como un checkbox junto al campo de envío de mensajes al chat que lo marcara como duda y una pestaña en la visualización del evento que permita recoger las dudas, de manera que el ponente (o usuario autorizado a ello), y solamente él, pudiera responderlas. En el ámbito de un curso e-Learning, este ponente podría ser el profesor.

Envío del evento mediante e-mail

Otra funcionalidad que podría ofrecerse sería la de enviar por email un resumen de contenidos del evento, esto es, un listado de los materiales, sus URLs y textos que los acompañan, así como los tuits generados con el hashtag del evento desde la fecha de inicio del evento hasta la fecha de la generación del resumen.

Creación de eventos vía web

Ya que la creación de nuevos eventos se basa en el consumo de servicios web, no sería especialmente difícil crear una interfaz web para la creación de los eventos. Esto haría infinitamente más cómoda y rápida la creación de nuevos eventos en la aplicación.

Incorporar un buscador de materiales

Otra mejora muy interesante pero no trivial podría ser utilizar las API's públicas de diferentes aplicaciones web para obtener los materiales. De esta manera, en la propia aplicación el usuario podría buscar videos alojados en Youtube, dejar que el usuario lo visualice y, si le conviene el material, seleccionarlo o marcarlo para

ser añadido. Lo mismo podría hacerse con las imágenes, por ejemplo, utilizando *Flickr*.

Inclusión automática de títulos y descripciones

Cuando un usuario añade un video o presentación y no añade título o descripción, estos datos podrían obtenerse de la información devuelta por el servicio oEmbed de youtube y de Slideshare, introduciéndolos como valores por defecto siempre y cuando el usuario no hubiera añadido esa información.

7. Conclusiones personales

Este proyecto me ha enseñado algo en varios aspectos: Desde lo complejo que es desarrollar una aplicación desde cero intentándolo hacer todo correctamente desde el principio hasta lo complejo que es decidirse por una tecnología concreta para iniciar el desarrollo de la aplicación, ya que rara vez hay una solución perfecta.

Inicialmente iba a desarrollarse la aplicación como una aplicación Web ya que profesionalmente me dedico al desarrollo Web y me parecía la opción más rápida y sencilla, pero opté por hacer un desarrollo nativo porque creíamos, tanto el tutor como yo, que era una oportunidad para aprender algo nuevo y útil, con salidas laborales y con creciente demanda.

Una vez desarrollado el proyecto me he dado cuenta de que no ha sido suficientemente grande o complejo para profundizar en el aprendizaje de la plataforma Android (debido mayormente a que los requerimientos de la aplicación no necesitaban características muy concretas de la plataforma) y que el resultado obtenido no es tan bueno como yo deseaba al comienzo del proyecto ni como seguramente hubiera obtenido desarrollando la aplicación como una Web, debido a la falta de conocimientos iniciales del sistema Android y sobre todo a la forma tan diferente que Android tiene de abordar la representación gráfica comparado con la manera en que lo hace un navegador Web.

Aún así, o tal vez por ello, ha sido una experiencia gratificante el ver la aplicación en funcionamiento después de los muchos problemas que han surgido en el desarrollo (como los cambios en la API de Twitter o la dificultad de adaptar las vistas de la aplicación a diferentes tamaños y densidades de pantalla).

Si hoy en día tuviera que rehacer la aplicación seguramente optaría por otras tecnologías, pero eso es algo que no sabría si no hubiera probado la solución escogida para el desarrollo de este proyecto.

8. Acrónimos

A continuación se recogen algunos acrónimos no descritos en el documento

- **AJAX:** Asynchronous JavaScript And XML
- **API:** Application Programming Interface
- **DOM:** Document Object Model
- **JSON:** JavaScript Object Notation
- **JVM:** Java Virtual Machine
- **REST:** Representational State Transfer
- **SDK:** Software Development Kit
- **XML:** Extended markup Language

9. Glosario de términos

A continuación se definen algunos conceptos utilizados en el documento y que podrían necesitar de una aclaración para algunos lectores.

AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DOM.

API

Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece una cierta biblioteca para ser utilizado por otro software.

Applet

Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo en un navegador web. El applet debe ejecutarse en un contenedor, que le proporciona un programa anfitrión.

Back end

En el diseño de software, el back end es la parte encargada de procesar la entrada recibida del usuario.

Callback

Es una función "A" que se usa como argumento de otra función "B". Cuando se llama a "B", ésta ejecuta "A". Para conseguirlo, usualmente lo que se pasa a "B" es el puntero a "A".

E-learning

Se denomina aprendizaje electrónico (conocido también por el anglicismo e-learning) a la educación a distancia completamente virtualizada a través de los nuevos canales electrónicos, en especial Internet, utilizando para ello herramientas o aplicaciones de hipertexto (v.g. correo electrónico, foros de discusión, mensajería instantánea) como soporte de los procesos de enseñanza y aprendizaje.

Framework

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Front end

En el diseño de software, el front end es la parte encargada de la interacción con los usuarios.

Hashtag

En la red social Twitter, un hashtag es una cadena de caracteres que empieza por el carácter “#” y que se utiliza para representar un tema de conversación.

JSON

Es un formato de notación de objetos JavaScript ampliamente utilizado en las comunicaciones entre aplicaciones como alternativa al lenguaje XML.

JVM

Una máquina virtual Java (en inglés Java Virtual Machine, JVM) es una máquina virtual capaz de interpretar y ejecutar en el sistema operativo instrucciones expresadas en un código binario especial, el cual es generado por el compilador del lenguaje Java.

Kernel

Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware del ordenador, es decir, es el encargado de gestionar recursos de la máquina. Con el objetivo de optimizar el uso de la máquina, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo.

Máquina virtual de java:

ver JVM

REST

Es una técnica de arquitectura software para sistemas distribuidos que permite, a través del protocolo HTTP, comunicar a dos sistemas, permitiéndoles compartir información e incluso recibir órdenes junto a los parámetros necesarios para llevarlas a cabo.

SDK

Es un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.

Smartphone

Es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad para almacenar datos y realizar actividades semejantes a una mini computadora y conectividad que un teléfono móvil convencional.

Twitcam:

Es una aplicación web que permite realizar streaming de video y que implica el uso de la cuenta de Twitter de un usuario.

Wireframe

En diseño de aplicaciones informáticas, un wireframe es una representación esquemática de la apariencia de la aplicación sin elementos gráficos que muestran contenido y comportamiento de la misma. Sirven como herramienta de comunicación y discusión entre arquitectos de información, programadores, diseñadores y clientes. También se pueden utilizar para comprobar la usabilidad de una aplicación.

Wizard

En el desarrollo de software, un wizard es un asistente mediante interfaz de usuario que presenta al usuario una secuencia de pantallas que le solicita decisiones o valores y que le asiste en la realización de una tarea.

XML

Es un lenguaje de marcado desarrollado por el W3C utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

10. Referencias

Estas son las URLs a las que se hace referencia durante el documento

[URL 1] [Último acceso: 24-October-2013]

<http://www.seroundtable.com/google-hangouts-15-15962.html>

[URL2] [Último acceso: 24-October-2013]

<http://www.slideshare.net/developers/apikits>

[URL3] [Último acceso: 24-October-2013]

<http://www.eventmobi.com/about/why-eventmobi/>

[URL4] [Último acceso: 24-October-2013]

<http://www.w3schools.com/tags/default.asp>

[URL5]] [Último acceso: 4-Marzo-2013]

http://www.w3schools.com/html5/html5_ref_globalattributes.asp

[URL6] [Último acceso: 24-October-2013]

http://www.w3schools.com/tags/att_input_type.asp

[URL7] [Último acceso: 24-October-2013]

<http://www.cristalab.com/tutoriales/validar-formularios-de-forma-nativa-con-html5-c1048771/>

[URL8] [Último acceso: 24-October-2013]

<http://www.cssblog.es/6-novedades-css3-que-puedes-implementar-hoy/>

[URL9] [Último acceso: 24-October-2013]

<http://www.html5rocks.com/en/tutorials/speed/html5/>

[URL10] [Último acceso: 24-October-2013]

http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

[URL12] [Último acceso: 24-October-2013]

<http://www.anieto2k.com/2006/09/06/selectores-css-que-deberias-conocer/>

[URL13] [Último acceso: 24-October-2013]

<http://es.wikipedia.org/wiki/JQuery>

[URL14] [Último acceso: 24-October-2013]

<http://jquerymobile.com/>

[URL15] [Último acceso: 24-October-2013]

<http://www.w3.org/TR/2011/WD-html5-20110525/elements.html#embedding-custom-non-visible-data-with-the-data-attributes>

[URL16] [Último acceso: 24-October-2013]
<http://mamoreno.com/palabras/words/listar>

[URL17] [Último acceso: 24-October-2013]
<http://noticias.universia.edu.ve/ciencia-nntt/noticia/2011/03/21/803091/conoce-cinco-navegadores-Web-mas-populares.html>

[URL18] [Último acceso: 24-October-2013]
<http://caniuse.com/#search=websockets>

[URL19] [Último acceso: 24-October-2013]
http://www.cincodias.com/articulo/empresas/penetracion-movil-espana-desploma-puntos-ano/20130117cdscdsemp_17/

[URL20] [Último acceso: 24-October-2013]
<http://www.nielsen.com/us/en/newswire/2012/smartphones-account-for-half-of-all-mobile-phones-dominate-new-phone-purchases-in-the-us.html>

[URL21] [Último acceso: 24-October-2013]
<http://bitelia.com/2012/05/android-es-el-sistema-operativo-del-485-de-los-smartphones-en-los-estados-unidos>

[URL22] [Último acceso: 24-October-2013]
<http://es.nielsen.com/site/index.shtml>

[URL23] [Último acceso: 24-October-2013]
http://es.wikipedia.org/wiki/Windows_Phone

[URL24] [Último acceso: 24-October-2013]
<http://www.windowsphone.com/es-es/features>

[URL25] [Último acceso: 24-October-2013]
<http://thenextweb.com/mobile/2013/02/25/nokia-ceo-there-are-now-130000-apps-available-for-windows-phone/>

[URL26] [Último acceso: 24-October-2013]
<http://bitelia.com/2012/05/android-es-el-sistema-operativo-del-485-de-los-smartphones-en-los-estados-unidos>

[URL27] [Último acceso: 24-October-2013]
<http://www.miguel diazrubio.com/2011/12/30/desarrollo-ios-tipos-de-licencias-de-desarrollo/>

[URL28] [Último acceso: 24-October-2013]

<http://www.tablerodeanálisis.es/2012/02/analisis-android-vs-iphone/>

[URL29]] [Último acceso: 12-Junio-2013]

<http://view.officeapps.live.com/op/view.aspx?src=http%3a%2f%2ffiles.ch9.ms%2fmix%2f2011%2fppt%2fdvc19.pptx>

[URL30] [Último acceso: 24-October-2013]

<http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>

[URL31] [Último acceso: 24-October-2013]

<http://upload.wikimedia.org/wikipedia/commons/thumb/e/e2/CocoaTouchLayers.png/626px-CocoaTouchLayers.png>

[URL32] [Último acceso: 24-October-2013]

<http://www.droidlife.com/>

[URL33] [Último acceso: 24-October-2013]

<http://www.xatakandroid.com/moviles-android/se-muestra-google-sooner-un-prototipo-de-2007-con-un-android-recien-nacido>

[URL34] [Último acceso: 24-October-2013]

http://es.wikipedia.org/wiki/Open_Handset_Alliance

[URL35] [Último acceso: 24-October-2013]

<http://www.xpertos.tv/tecnologia/Articulos/VistadeArt%C3%ADculos/tabid/221/Article/749/versiones-de-android-las-conocas-todas.aspx>

[URL36] [Último acceso: 24-October-2013]

<http://www.techradar.com/news/phone-and-communications/mobile-phones/android-5-0-key-lime-pie-release-date-news-and-rumours-1091500>

[URL37] [Último acceso: 24-October-2013]

<http://developer.android.com/about/dashboards/index.html>

[URL38] [Último acceso: 24-October-2013]

<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

[URL39] [Último acceso: 24-October-2013]

<http://androideity.com/wp-content/uploads/2011/07/arquitecturaandroid.png>

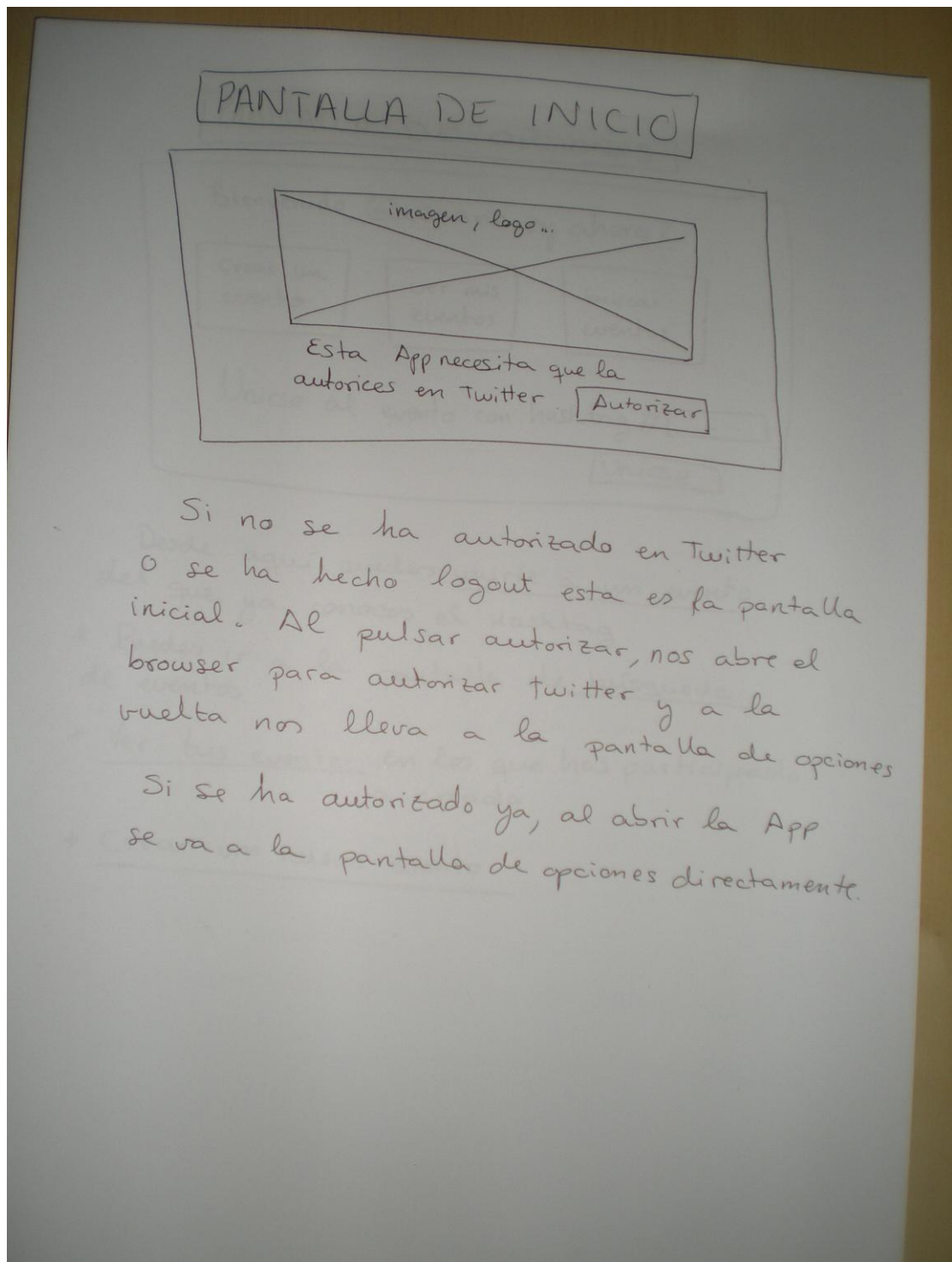
[URL40] [Último acceso: 24-October-2013]

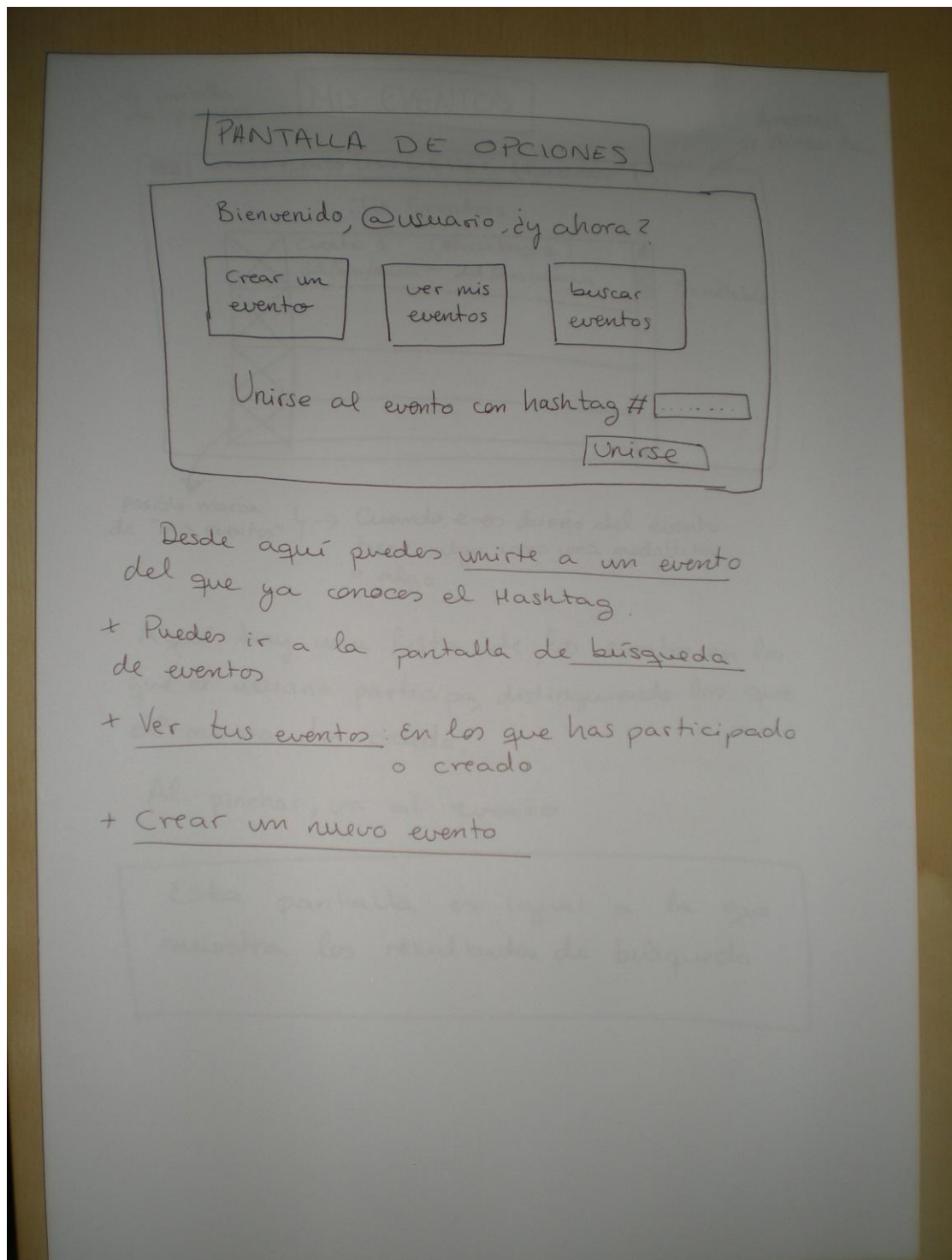
<http://androideity.com/wp-content/uploads/2011/07/system-architecture.jpg>

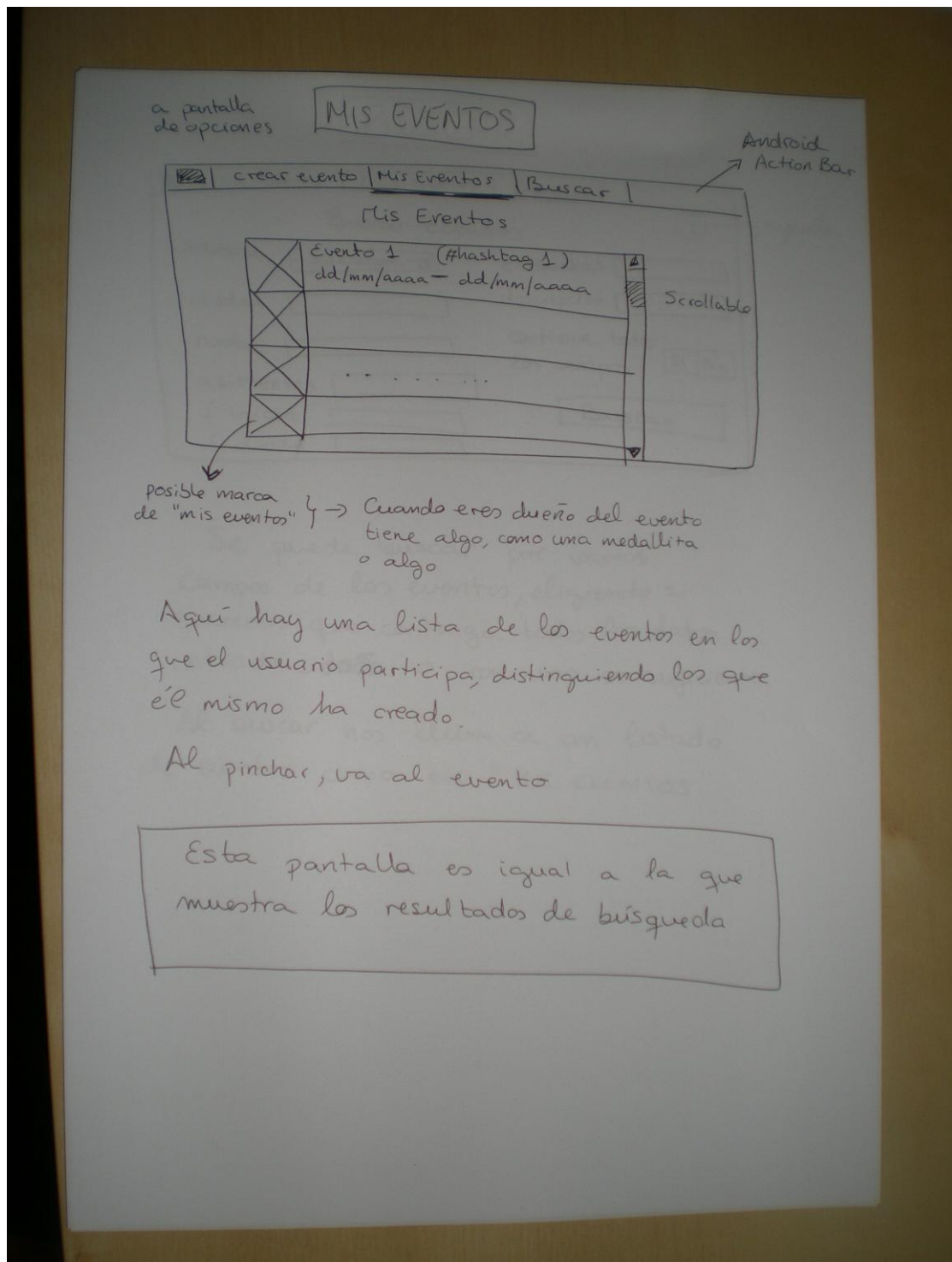
- [URL41] [Último acceso: 24-Octubre-2013]
<http://developer.android.com/guide/components/fundamentals.html>
- [URL42] [Último acceso: 24-Octubre-2013]
http://www.androidjavadoc.com/1.0_r1_src/android/app/doc-files/activity_lifecycle.png
- [URL43] [Último acceso: 24-Octubre-2013]
<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-conceptos-generales-1>
- [URL44] [Último acceso: 24-Octubre-2013]
<http://www.sqlite.org/about.html>
- [URL45] [Último acceso: 24-Octubre-2013]
<http://www.somoslibres.org/modules.php?name=News&file=article&sid=183>
- [URL46] [Último acceso: 24-Octubre-2013]
<http://www.ch-werner.de/sqliteodbc/>
- [URL47] [Último acceso: 24-Octubre-2013]
<https://sites.google.com/site/swcuc3m/home/android/generalidades/dalvikvm-1>
- [URL48] [Último acceso: 24-Octubre-2013]
http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software
- [URL49] [Último acceso: 24-Octubre-2013]
http://upload.wikimedia.org/wikipedia/commons/3/3c/Esquema_general_de_una_metodologia_agil_para_desarrollo_de_software.png
- [URL50] [Último acceso: 24-Octubre-2013]
http://www.volere.co.uk/pdf%20files/template_es.pdf
- [URL51] [Último acceso: 24-Octubre-2013]
<http://www.genbetadev.com/frameworks/twitter4j-integracion-de-tu-aplicacion-java-con-twitter>
- [URL52] [Último acceso: 24-Octubre-2013]
<http://www.apache.org/licenses/LICENSE-2.0>
- [URL53] [Último acceso: 24-Octubre-2013]
<http://es.wikipedia.org/wiki/OAuth>
- [URL54] [Último acceso: 24-Octubre-2013]
<https://dev.twitter.com/docs/api/1.1/get/search/tweets>
- [URL55] [Último acceso: 24-Octubre-2013]
<http://developer.android.com/training/displaying-bitmaps/cache-bitmap.html>

11. ANEXO A

**Imágenes de los *wireframes* iniciales
desarrollados a mano**







BUSCAR

☒ crear evento
Mis eventos
Buscar

Buscar Eventos

(?) ayuda

Idioma -Selecciona-

ciudad

nombre

asistentes

f.inicio

f.desde

etiquetas

descripcion

contiene todos los campos Sí No

Buscar

Se puede buscar por varios campos de los eventos, eligiendo si queremos que contenga todos los datos introducidos ~~dos~~ o con uno es suficiente.

Al buscar nos lleva a un listado de eventos como en "MIS EVENTOS"

CREAR EVENTO

crear evento
mis eventos
Buscar

Nombre

Idioma

Hashtag #

f. inicio

f. fin

Ciudad

etiquetas

Twitear creación del evento ☐ Sí ☐ No

Notificar creación a

Descripción del evento

Siguiente

crear evento
mis eventos
Buscar

Añadir materiales

Slideshare

Youtube

Imagen*

es lista? ☐ Sí ☐ No

ANTERIOR
TERMINAR

La creación irá en pantallas, como un "wizard". Podría ser una pantalla por cada tipo de material, introduciendo material y un texto descriptivo. De momento un solo material de cada tipo

